UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

Diarización de locutores en señales de audio de radiotelevisión

Pablo Ramírez Hereza
Tutor: Javier Franco Pedroso
Ponente: Joaquín González Rodríguez

Junio 2017

Diarización de locutores en señales de audio en radiotelevisión

AUTOR: Pablo Ramírez Hereza
TUTOR: Javier Franco Pedroso

Área de Tratamiento de Voz y Señales (ATVS)
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2017

Resumen

Este trabajo de fin de grado tiene como principal objetivo la implementación y el análisis de diferentes técnicas utilizadas para el desarrollo de un sistema de diarización de locutores en el contexto de la evaluación Albayzín 2016 de diarización de locutores.

La diarización de locutores consiste en, dado un audio de entrada determinar los intervalos de tiempo en los que intervienen distintos locutores sin tener ningún tipo de información adicional, además del propio audio, de forma independiente al tipo de canal y a la presencia de cualquier tipo de ruido de fondo.

El desarrollo de este trabajo se divide en dos etapas. La primera, condicionada por la evaluación Albayzín, desarrollada entre los meses de septiembre y octubre de 2016, se centra en el acondicionamiento de un sistema de referencia a los datos de entrenamiento y desarrollo proporcionados, y en el análisis de técnicas alternativas a dicho sistema de referencia con el objetivo de mejorar su rendimiento.

Por otra parte, la segunda etapa se centra en la incorporación de técnicas basadas en i-vectors a etapas específicas del proceso de diarización, desarrollando un nuevo sistema y midiendo su rendimiento en las mismas condiciones de la evaluación Albayzín 2016.

De esta forma, este trabajo de fin de grado, en el contexto de la evaluación Albayzín 2016, nos permitirá estudiar varias técnicas utilizadas en la actualidad en las distintas etapas de un sistema de diarización: extracción de características, detección de actividad, segmentación y agrupamiento. Además, nos proporcionará una comparativa entre ellas en términos de rendimiento y de tiempo de ejecución que conlleva cada una.

Palabras clave

Audio, procesado, diarización, clustering, i vectors, segmentación, características, detector, actividad, i-vectors.

Abstract

The main goal of this bachelor degree thesis is to implement and analyze different techniques that allow to develop a speaker diarization system in the context of the Albayzin 2016's speaker diarization evaluation.

Speaker diarization consist in determining the time intervals in which different speakers are taking part in a given recording, without any additional information besides the audio signal, where different transmission channel characteristics or background noise may appear.

This work can be divided into two different parts. The first one, conditioned by the Albayzin evaluation, is focused on adapting our reference system to the training and development data provided for the evaluation and using alternative techniques to improve our system performance.

On the other hand, the second part is focused on incorporating i-vectors-based techniques to specific stages of the speaker diarization process, developing a new system and measuring its performance in the same conditions of defined by the Albayzin 2016 evaluation.

Thus, this bachelor degree thesis, using the Albayzin evaluation as our general frameworks, will allow us to study some techniques commonly used in different stages of a speaker diarization system: features extraction, activity detection, segmentation and clustering. Furthermore, it will provide us a comparative analysis between these different techniques in terms of performance and execution time for each one.

Keywords

Audio, processing, diarization, clustering, i-vectors, segmentation, features, detector, activity, i-vectors.

Agradecimientos

Toda etapa tiene un fin. Este Trabajo de fin de grado supone el adiós a una etapa de mi vida en la cual he crecido, no solo intelectualmente sino también como persona. Tras 5 duros años de esfuerzo, constancia y muchas transformadas de Fourier, ya solo queda dar las gracias. Primero agradecer a la Escuela Politécnica Superior el abrirme sus puertas y enseñarme lo que ahora sé con certeza que es mi vocación. Gracias a todo el personal docente que ha sabido contagiarme, además de sus conocimientos, su ilusión y su pasión por esta carrera. Mención de honor a profesores como Doroteo Torre, Joaquín Rodríguez, Daniel Ramos, Jorge Ruíz y Álvaro García que hacen que estudiar esta carrera en este centro sea único.

Agradecer también a mi ponente Joaquín Rodríguez, por permitirme acabar esta etapa con un trabajo especial, al que ha sido un placer dedicarle mucho empeño y tiempo. A mi tutor, y mi compañero a lo largo de este final, Javier Franco, por su incansable paciencia, por sus continuas explicaciones y, sobre todo, por su constante interés en mí, sé que serás un gran profesor.

No puedo olvidarme de los grandes culpables de que estos 5 años hayan sido tan rápidos, tan intensos y hayan parecido una especie de tragicomedia. La mayor alegría de esta etapa no ha sido aprobar ondas sino encontrar amigos como Miguel Basarte, Emilio Gómez, Sergio Serra, José Gil y Sergio Cortés. Sois únicos y aquí tenéis a un fiel amigo para siempre.

Por último, agradecer a mi pequeña, por ser el gran apoyo sin el cual todo hubiese sido más complicado, por aguantarme todos los días y hacerme reír hasta en los más difíciles momentos. A mi familia, por creer siempre en mí y en mis capacidades, y por nunca dejar de protegerme incluso de mí mismo. Para terminar, agradecer a mi abuelo Juan por enseñarme que rendirse nunca es una opción y que la mejor arma de un hombre es su razonamiento.

Pablo Ramírez Hereza
Mayo 2016

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

1.1 Motivación

El continuo desarrollo de las tecnologías de la información, el aumento de la capacidad de almacenamiento y del ancho de banda, además de la generalización del acceso y depósito de grandes cantidades de contenido multimedia, son las razones por las cuales, en las últimas décadas, aparece la necesidad de la aplicación de tecnologías automáticas que permitan una búsqueda rápida y eficiente de contenido de audio.

En este contexto aparece la segmentación de audio y, más concretamente, la diarización de locutores, que es el objeto de este trabajo. Se puede definir la "diarización de locutores" como el proceso automático consistente en el particionado de una señal de entrada en segmentos homogéneos en función de un etiquetado de cada locutor sin tener ningún tipo de información previa sobre el contenido del audio, como son el número e identidad de los intervinientes, el idioma empleado, el posicionamiento de los micrófonos o información acerca del ruido y/o de la música de fondo.

La carencia de cualquier tipo de información adicional al audio bajo análisis hace que la automatización de este proceso sea complicada. Además, los sistemas de diarización de locutores deben enfrentarse a los problemas típicos de cualquier sistema de procesado de audio, como son la presencia de ruido de fondo, el solapamiento de varios locutores, las características de la sala, el cambio de emociones o idiomas de los locutores, la variabilidad intra-locutor y las limitaciones y condiciones propias de cada canal.

Como ha sido mencionado anteriormente, el objetivo principal de un sistema de diarización es, a partir de un audio de entrada dado, responder a la pregunta *¿Cómo se distribuye el tiempo de una grabación entre los diferentes locutores?* Pese a que las respuestas a esta pregunta, por sí solas, contengan información de gran utilidad para la generación de metadatos para el indexado y para el análisis de grabaciones de audio, como pueden ser la clasificación del tiempo y del número de participaciones de los intervinientes en un debate o en una entrevista, otra función de gran importancia de todo sistema de diarización de locutores es servir como etapa de preprocesamiento previa para otros sistemas de procesado de audio. La **Figura 1** representa un ejemplo de cómo se integraría un sistema de diarización con otros sistemas. Los sistemas de diarización pretenden mejorar el rendimiento de sistemas de reconocimiento de voz, de sistemas de transcripción automática, de reconocimiento de emociones y, en general, de cualquier sistema de procesado de audio gracias a su adaptación a cada uno de los intervinientes. Pero, además, también representa una etapa imprescindible para procesos como el reconocimiento de locutores.

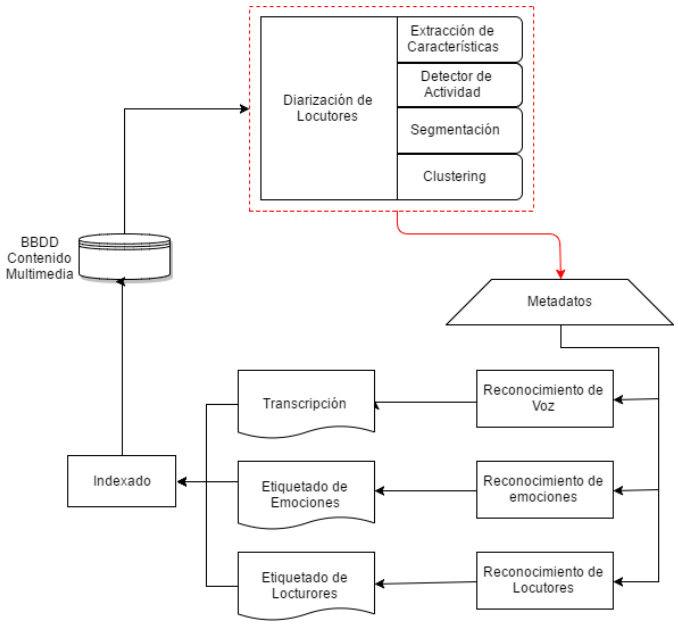


Figura 1. Diagrama de bloques diarización como pre-procesado de audio.

La existencia de evaluaciones de diarización, como la Albayzín, es prueba del interés y la demanda de este tipo de sistemas para las aplicaciones mencionadas (1). De este modo, este trabajo de fin de grado, mediante la comparación de diferentes sistemas en el contexto de la evaluación Albayzin 2016 en Diarización de locutores, aporta una vista panorámica de algunas técnicas actualmente utilizadas, permitiendo comprobar sus ventajas y limitaciones.

1.2 Objetivos

Los objetivos fijados para este trabajo son los siguientes:

Primero, realizar un estudio de algunas de las técnicas utilizadas en la actualidad en distintas etapas de un sistema de diarización.

Por otro lado, ajustar un sistema de referencia ya desarrollado y descrito en (2) para participar en la evaluación Albayzín 2016 de diarización de locutores.

Y, por último, implementar técnicas alternativas en algunas de las etapas de la diarización y comparar el rendimiento de los sistemas alternativos con el sistema de referencia en el contexto de la evaluación.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1: Introducción**

  Contiene la motivación y los objetivos de este proyecto.

- **Capítulo 2: Estado del arte**

  En este capítulo podemos encontrar una descripción de la estructura general de un sistema de diarización y de algunas de las técnicas utilizadas en cada una de las etapas que lo conforman.

- **Capítulo 3: Diseño**

  En este apartado veremos un resumen de las características que definen a los sistemas finales estudiados o implementados sobre los cuales se ha trabajado a lo largo de este documento.

- **Capítulo 4: Desarrollo**

  Una vez definido el marco teórico, y definido las principales características de nuestros sistemas finales, esta sección describe el proceso realizado hasta la obtención de resultados, además de una descripción del entorno experimental, de las herramientas utilizadas, de los datos sobre los que se realiza la investigación y de la propia evaluación Albayzín 2016.

- **Capítulo 5: Integración, pruebas y resultados**

  Tras explicar todas las pruebas realizadas, este capítulo nos aportará un desglose de los resultados obtenidos por cada sistema para así poder compararlos.

- **Capítulo 6: Conclusión y trabajo futuro**

  Por último, se expondrán las conclusiones del trabajo realizado, además de las posibles pruebas y mejorías que se pueden realizar en un futuro en nuestro sistema.

2 Estado del arte

En este capítulo describiremos la arquitectura más habitual de un sistema de diarización y repasaremos algunas de las técnicas más utilizadas para cada una de las etapas involucradas.

2.1 Arquitectura

La diarización de locutores realiza un particionado de una señal de audio de entrada en diferentes segmentos. Aunque se desconoce la identidad real del locutor, la diarización separa los segmentos donde hablan distintas personas y se identifican aquellos segmentos que corresponden a la misma persona. De esta forma, y sin ningún tipo de procesado adicional, como el reconocimiento de la identidad de los locutores que intervienen, el sistema de diarización tiene como salida un etiquetado de la señal del audio en función de los diferentes intervinientes sin aportar su identidad real, este etiquetado de la señal se encuentra descrito en el Apartado 4.3.

Para obtener dicho resultado, la mayoría de las aproximaciones siguen la siguiente arquitectura, la misma que nosotros analizaremos e implementaremos a lo largo de este trabajo:

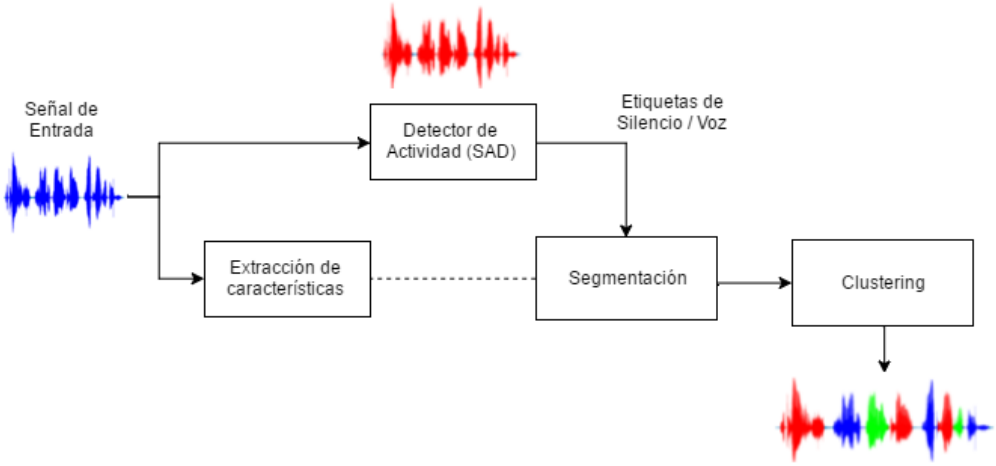


Figura 2. Arquitectura típica de un sistema de diarización

Como podemos ver en la **Figura 2**, la primera etapa del sistema de diarización es la **extracción de características** propias de cada una de los intervinientes que nos permiten una distinción entre ellos. Paralelamente, o posteriormente en algunos casos, encontramos el **detector de actividad**. Este módulo cataloga el audio de entrada en segmentos con y sin voz. Esto permite prescindir de los segmentos sin voz del análisis posterior. La **segmentación** es el proceso encargado de detectar los puntos de cambio de locutor en las características extraídas de la señal de audio. Por último, **el agrupamiento** es la fase encargada de asignar a cada segmento una etiqueta de identidad. Para ello, se realiza una agrupación de segmentos en función de una medida de distancia entre sus características.

2.2 Extracción de características

2.2.1 Mel Frequency Cepstral Coefficients (MFCCs)

Cada locutor contiene en su voz una serie de características propias que lo distinguen del resto derivadas de la fisionomía particular del tracto vocal y los hábitos de pronunciación y articulación propios de cada persona. Como en todo sistema de reconocimiento de patrones, el primer paso del sistema de diarización es extraer estas características que nos permiten distinguir unos locutores de otros. Las características más utilizadas son los Coeficientes Cepstrales Lineales o *Linear Frequency Cepstral coefficients (LFCC)*, los coeficientes predictivos lineales o *Linear Predictive Coefficients (LPC)* y, en especial, los Coeficientes cepstrales o *Mel-Frequency Cepstral Coefficients (MFCCs)*. En este trabajo nos centraremos en los coeficientes MFCCs.

El proceso de extracción de MFCCs puede ser resumida en los siguientes pasos como se ve reflejada en la **Figura 3**:

- La señal de audio es analizada en intervalos de tiempo cortos (tramas). Normalmente para cada audio se usan tramas de 20 ms con un solapamiento del 50% entre tramas consecutivas.

- Cada trama de señal es enventanada, normalmente con una ventana Hamming.

- Después, se obtiene el espectro de frecuencias de cada trama enventanada mediante la Fast Fourier Transform (FFT).

- Posteriormente se le aplica a la señal un banco de filtros triangulares en frecuencia como muestra la figura. Estos filtros se encuentran equiespaciados en el eje de frecuencias Mel que sigue una escala perceptual.

- Se calcula la energía logarítmica a la salida de cada filtro.

- Por último, se aplica la Transformada discreta del coseno o DCT para agrupar la información en el menor número de coeficientes posibles.

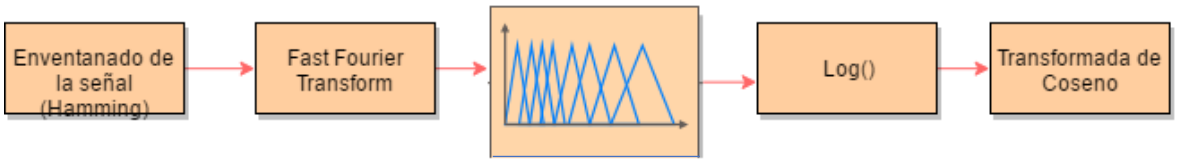


Figura 3. Diagrama de bloques del cálculo de coeficientes MFCCs

2.2.2 Normalización de Canal

La principal limitación de todas estas características es que representan, no solo la información propia de cada locutor, también la información adicional presente en el canal de comunicación ya sea una conversación telefónica, el ruido debido a las reverberaciones en una sala o simplemente ruido de fondo.

Es por esto que se suelen aplicar técnicas de normalización de canal para así intentar minimizar el impacto de éste y para que los coeficientes MFCC reflejen únicamente las características del locutor. En el dominio cepstral, el efecto del canal se convierte en un término aditivo a la señal, por lo que este efecto puede estimarse a partir de la media y eliminarse simplemente mediante su resta. Dos ejemplos de normalización de canal son la *Cepstral Mean Normalization (CMN)* y la *Cepstral Mean and Variance Normalization (CMVN).*

Estas dos técnicas son utilizadas principalmente por su sencillez. Denominemos a la media μ y la varianza σ de los coeficientes MFCCs $c(t)$ definidas en las siguientes fórmulas siendo N el número de tramas.

$$\mu = \frac{1}{N} * \sum_t c(t) \quad (2.1)$$

$$\sigma^2 = \frac{1}{N} * \sum_t (c(t) - \mu)^2 \quad (2.2)$$

Una vez calculadas media y varianza, podemos calcular sencillamente los coeficientes normalizados mediante CMN y CMVN de la siguiente forma respectivamente:

$$c_{norm-CMN}(t) = c(t) - \mu \quad (2.3)$$

$$c_{norm-CMVN}(t) = \frac{c(t)-\mu}{\sigma} \quad (2.4)$$

2.2.3 Coeficientes derivativos

Los MFCCs, calculados tal y como se describe en el Apartado 2.2.1, reflejan la información espectral estática presente en la voz. Es habitual (3) añadir a dichos coeficientes información adicional dinámica que refleje la variación de estas características a lo largo del tiempo.

Esta información se puede obtener añadiendo a los MFCCs los coeficientes cepstrales derivativos de primer orden $\Delta c(t)$ y de segundo orden $\Delta\Delta c(t)$.

Los coeficientes de primer orden pueden calcularse de la siguiente forma:

$$\Delta c(t) = \frac{\sum_{n=1}^{N} n(c(t+n) - c(t-n))}{2\sum_{n=1}^{N} n^2} \quad (2.5)$$

Siendo *N* el número de tramas antes y después, sobre los que se calcula la derivada discreta que representa (2.5). Mientras que los coeficientes derivativos de segundo grado se obtienen aplicando la fórmula anterior a los coeficientes derivativos de primer grado.

2.3 Speech Activity Detector/ Detector de Actividad (SAD)

El objetivo de un sistema de diarización consiste en determinar los segmentos en los que hablan distintos locutores. Sin embargo, en una grabación pueden aparecer, además de tramos de silencio, otros eventos acústicos como por ejemplo ruido o música de fondo, que no interesa procesar o pueden dar lugar a errores de etiquetado. Esto supone la necesidad de un detector de actividad que realice un correcto etiquetado de la señal de entrada en segmentos con y sin voz. El rendimiento del detector de actividad es determinante para el rendimiento total del sistema de diarización al ser directamente responsable de los errores de Falsos positivos (False Alarm Apartado 4.1.4) y Falsos negativos (Missed Speech Apartado 4.1.4).

2.3.1 Estrategias basadas en energía

La estrategia más intuitiva para el desarrollo de un detector de actividad se basa en la definición de un umbral de energía, según el cual, tramas que superen dicho umbral se pueden definir como segmentos de voz y, en el caso contrario, las tramas con menor energía que el umbral serán catalogadas como silencio.

Esta aproximación solo es efectiva en grabaciones donde solo aparezca voz sin ruido de fondo, pero no para grabaciones de radiodifusión en las que pueden aparecer otros eventos acústicos. Por lo tanto, no soluciona la principal complicación de esta etapa que reside en definir, no solo los segmentos de silencio como tal, sino que también tiene que añadir a esta categoría segmentos con música, ruido, aplausos o risas que en ocasiones pueden superar la energía de umbral.

2.3.2 Estrategias basadas en modelos

Una estrategia más elaborada consiste en el uso de modelos de mezclas de gaussianas (Gaussian Mixture Models GMMs) para representar los diferentes tipos de audio que podemos encontrar como por ejemplo silencio, voz, música, ruido, aplausos, voz por encima de música o voz por encima de ruido. De esta forma, se calcula la verosimilitud de cada trama de audio para cada uno de los modelos, y se escoge el que mayor probabilidad tenga.

Un ejemplo es la estrategia de detección de actividad del sistema LIUM (3). En el caso de LIUM el audio se asigna a uno de 8 posibles modelos, cada uno representado por un GMM diagonal de 64 componentes: 2 modelos de silencio -banda ancha y estrecha-, 3 modelos para banda ancha- voz, voz con ruido y voz con música-, 1 modelo de voz banda estrecha -voz telefónica-, 1 modelo para audios enlatados y un modelo para música.

2.3.3 Estrategias basadas en la trayectoria de los armónicos

La técnica de detección de actividad de este trabajo de fin de grado está basada en un análisis espectral del audio de entrada, en particular de la posición de los armónicos en un espectrograma (4).

La principal ventaja de esta técnica con respecto a la estrategia descrita en la sección 2.3.2 es que no necesita un banco de datos externo. Esta aproximación se basa en que, como podemos ver en la siguiente figura, una señal de voz presenta patrones en la posición de algunos armónicos debido al tracto vocal. Los armónicos de una señal de voz se mantienen breves espacios de tiempo a lo largo de los cuales varían ligeramente en frecuencia haciendo que tenga una trayectoria curva.

Esta característica permite también la diferenciación entre segmentos de música y de ruido: el espectrograma de un segmento musical se caracteriza por armónicos sostenidos durante periodos de tiempo relativamente largos, mientras que el de los segmentos de ruido se caracterizan por su aleatoriedad en frecuencia sin ningún patrón definido, como podemos ver en la **Figura 4**.

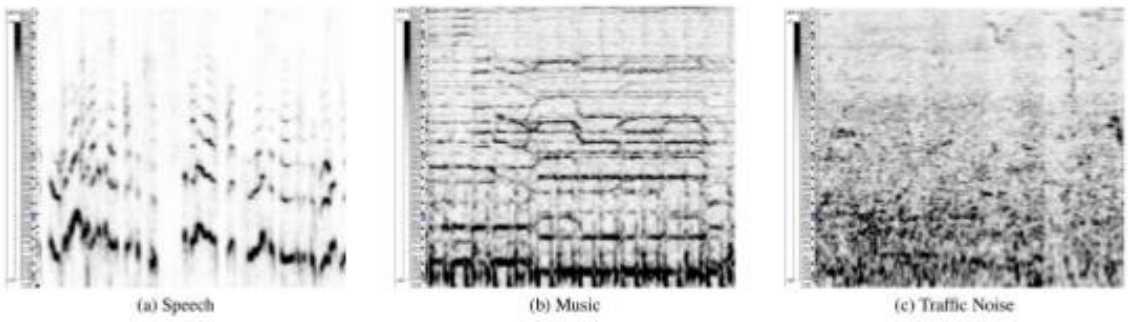


Figura 4. Espectrogramas de segmentos de voz, música y ruido.

.
De esta forma, la detección de voz se puede realizar mediante el análisis de la trayectoria de los armónicos. Para realizar este análisis de la trayectoria, tenemos que comparar el espectro de dos tramas de audio consecutivas X_t y $X_{t+offset}$.

Para ello, esta técnica obtiene el espectrograma logarítmico de la señal de entrada, sobre el cual se calcula la correlación cruzada definida en (2.6) donde l es el desplazamiento frecuencial sobre el que se calcula la correlación.

$$R_{xy}(l) = \sum_i X_t X_{t+offset}(l) \quad (2.6)$$

Para definir un valor que determine si hay o no voz, se define el valor $R = r_{xcorr} - r$ siendo r_{xcorr} el valor máximo de correlación y r el valor de la correlación cruzada para un desplazamiento frecuencial nulo. De esta forma, para señales de música, al tener patrones

espectrales horizontales, el valor máximo de correlación entre tramas conjuntas será cuando el desplazamiento frecuencial es nulo, por lo que, si desarrollamos el valor de R:

$$R = r_{xcorr} - r = R(0) - r = 0$$

Por lo contrario, para segmentos de voz, al ser las trayectorias de sus armónicas curvas y poco constantes, el valor máximo de correlación se produce con un determinado desfase frecuencial, *l*, no nulo y por lo tanto R > 0.

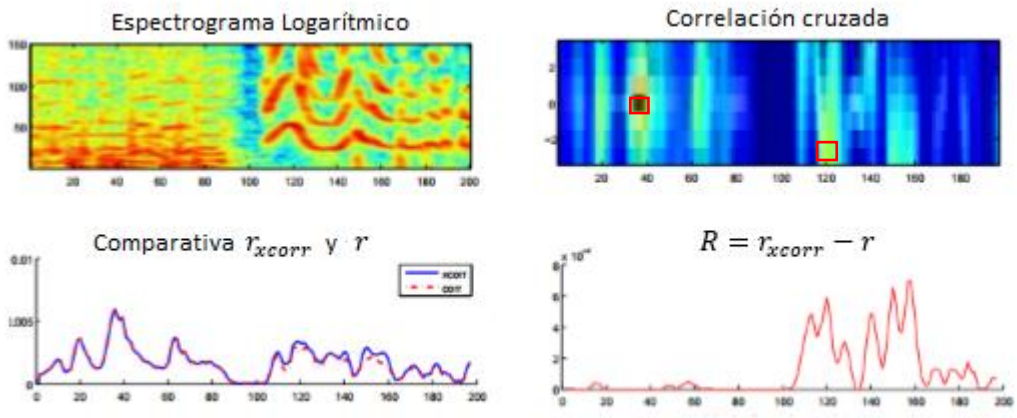


Figura 5. $R = r_{xcorr} - r$ para detección de actividad (2).

En la **Figura 5**, tenemos el proceso anteriormente explicado para una trama de audio en la que encontramos primero música y posteriormente solo voz. Podemos observar que, como quedaba representado en la **Figura 4,** la trayectoria de los armónicos del segmento de música es constante y la del segmento de voz es curva. Además, el máximo de la correlación cruzada es claramente para un desfase frecuencial nulo al contrario para el tramo de voz. Por último, esta figura nos muestra perfectamente como el valor de R es mayor que 0 para el tramo de voz mientras que para la música es casi nulo. Solo quedaría fijar un umbral de decisión que distinga los tramos hablados de los que no en función de este score.

2.4 Segmentación

La segmentación es una de las etapas clave de un sistema de diarización. Este proceso tiene como objetivo la detección de puntos de cambio de locutor en la señal de entrada o, en el caso de que la entrada de la segmentación provenga de un módulo de detector de actividad, solamente de los segmentos con voz.

Como queda reflejado en la **Figura 6**, la aproximación más habitual se basa en la comprobación de una hipótesis mediante una pareja de ventanas deslizantes. Para cada posición de las ventanas hay dos posibles hipótesis. La primera es que las características de las dos ventanas provienen de dos locutores diferentes y, por lo tanto, son representados por diferentes modelos.

Por el contrario, la segunda hipótesis se basa en que las características de ambas ventanas provienen del mismo locutor y pueden ser representadas por el mismo modelo.

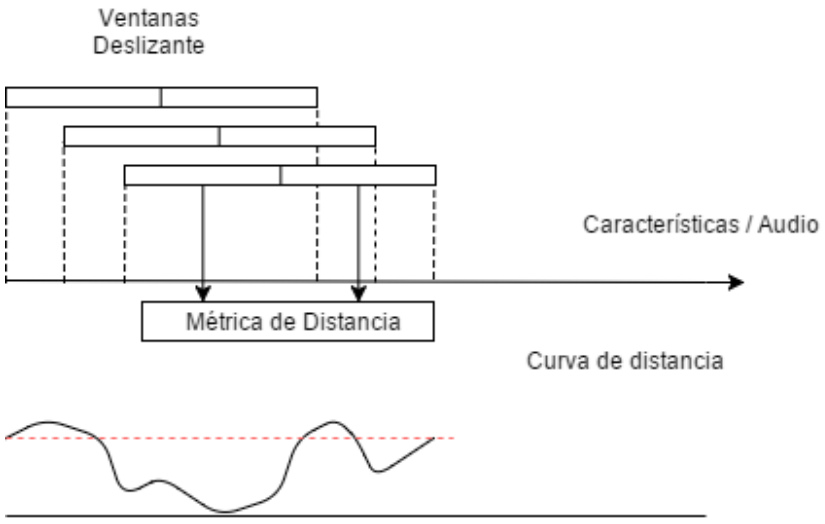


Figura 6. Aproximación a la segmentación

El resultado de la comparación de ambas hipótesis (medida de distancia entre distribuciones, ratio de verosimilitudes, etc.) da lugar una curva de distancia para cada posición de la ventana deslizante, cuyos máximos locales que superen un umbral fijado determinarán los puntos de cambio de locutor. Las técnicas más utilizadas para este propósito son la distancia basada en el criterio de información Bayesiana (*Bayesian Information Criteria BIC)* y el ratio de verosimilitud generalizado (*Generalized Likelihood Ratio GLR).*

Este umbral determinará la pureza de los segmentos finales, de tal forma que, si es muy bajo, se producirá una sobresegmentación que resultará en segmentos cortos de gran pureza. Si el umbral es muy alto, se detectará un menor número de puntos de cambio de locutor, lo que supone segmentos finales más largos, pero menos puros.

En la práctica, se busca una sobresegmentación antes que una escasa segmentación para asegurarse que no se pierde ningún punto de cambio, y eliminar todos aquellos puntos que en realidad no corresponden a un cambio mediante una etapa de agrupamiento posterior.

2.4.1 Distancia basada en el criterio de Información Bayesiana ΔBIC

La medida ΔBIC calcula la distancia entre dos conjuntos de características, c_i y c_j, para decidir si se modelan mejor con una misma distribución gaussiana o con dos diferentes de la siguiente forma:

$$\Delta\mathrm{BIC}(c_i||c_j) = (n_i + nj)\ln|\Sigma| - n_i \ln|\Sigma_i| - n_j \ln|\Sigma_j| - \lambda P \quad (2.7)$$

Donde Σ es la matriz de covarianza del conjunto unificado c (unión de $c_i\ y\ c_j$), Σ_i del conjunto c_i, Σ_j del conjunto c_j y $n_i\ y\ nj$ el número de vectores de características en los respectivos conjuntos. Denominamos P la penalización:

$$P = \frac{1}{2}(d + \frac{1}{2}d(d+1)) \ln n \quad (2.8)$$

Siendo d la dimensión de los vectores de características y n = $n_i + nj$.

El parámetro λ es utilizado para ajustar la penalización y determina el umbral de decisión del ΔBIC. Es por esto que dicho parámetro tendrá que ser ajustado a los datos.

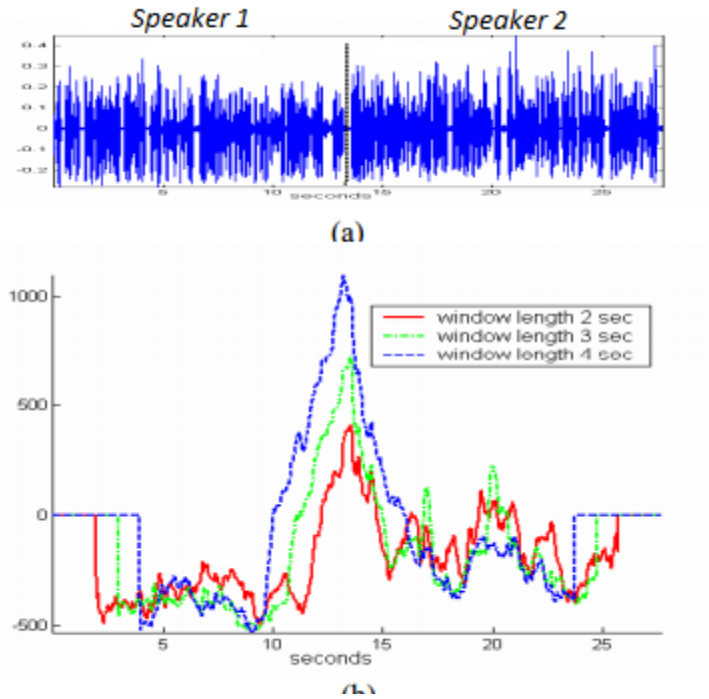


Figura 7. Ejemplo de audio de entrada (a) con su ΔBIC (b) en función de la longitud de las ventanas (18)

En la **Figura 7** podemos ver un ejemplo de la curva ΔBIC para un fragmento de audio en el que se localizan dos locutores diferentes. Podemos ver como la ΔBIC nos permite distinguir de forma eficaz los puntos de cambio de locutor si ajustamos correctamente un tamaño de ventana de análisis y un factor de penalización válido.

2.4.2 Generalized Likelihood Ratio (GLR)

Esta medida es un ratio de verosimilitudes que corresponde al cociente entre la probabilidad de un conjunto de características dada la hipótesis de que provienen del mismo locutor (H_1). y la probabilidad dada la hipótesis que provienen de diferente locutor (H_0)

$$GLR(c_i||c_j) = \frac{P(c|\mu,\Sigma)}{P(c_i|\mu_i,\Sigma_i)P(c_j|\mu_j,\Sigma_j)} = \frac{P(H_1)}{P(H_0)} \quad (2.9)$$

De forma similar a ΔBIC, se asume que para H_1 las características en el conjunto c (unión de c_i y c_j) se distribuyen de acuerdo a una única distribución gaussiana ($\{\mu, \Sigma\}$), mientras que para H_0 se considera una gaussiana diferente para cada subconjunto ($\{\mu_i, \Sigma_i\}$, $\{\mu_j, \Sigma_j\}$).

2.5 Agrupamiento (clustering)

Tras el proceso de segmentación, en el cual se encuentran los puntos de cambio entre locutores, es necesario identificar qué segmentos corresponden al mismo locutor y de cuales a diferentes locutores. Esta tarea se realiza en la etapa de agrupamiento o *clustering.*

2.5.1 Estrategias de agrupamiento

Como podemos ver en la **Figura 8**, encontramos dos estrategias principales para realizar la tarea de agrupamiento:

Por una parte, tenemos la estrategia *bottom-up*, también conocida como *Clustering Jerárquico Aglomerativo (AHC)*. Este esquema se basa en la sobresegmentación de una señal de audio en un número de segmentos mayor al número real de locutores. La primera etapa de este clustering consiste en considerar cada segmento un clúster diferente, posteriormente estos grupos iniciales se combinan gradualmente en función de una medida de distancia hasta que se cumple un criterio de parada y que se consigue un número deseado de clusters o locutores.

Por otro lado, tenemos la estrategia *top-down*. Al contrario del Clustering Jerárquico Aglomerativo, esta arquitectura parte de una baja segmentación de la señal de forma que el reducido número de clusters iniciales se van dividiendo en función del cumplimiento de una condición hasta conseguir un número de clusters deseados (2).

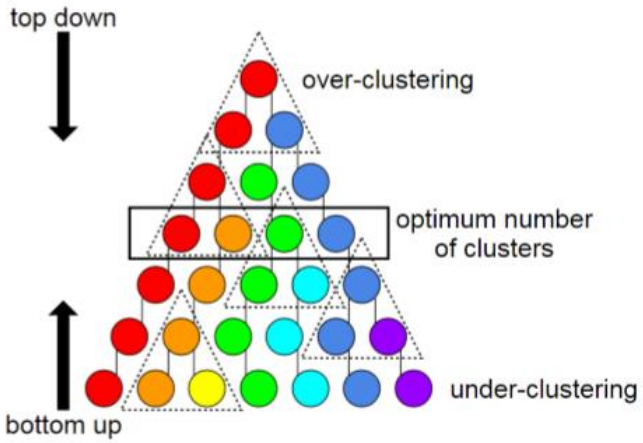


Figura 8. Aproximaciones Top-down y bottom-up en la que cada círculo representa un clúster y las líneas indican la fusión de dos clusters en uno o la división de uno en dos (28)

2.5.2 Clustering Jerárquico Aglomerativo (AHC)

La segmentación se puede considerar como una etapa de iniciación al clustering cuya salida es una sobresegmentación del audio de entrada en la que cada segmento forma un *cluster* diferente.

Para compensar esta sobresegmentación se aplica un clustering jerárquico Aglomerativo. Este proceso se puede dividir en los siguientes pasos.

1. Primero se comparan los diferentes *clusters* por pares mediante una medida de distancia (GLR Apartado 2.4.2 o ΔBIC Apartado 2.4.1, por ejemplo).

2. Mediante el resultado de esta medida de distancia, y la definición de un umbral de decisión, se unen o no las parejas de *clusters*.

3. Rehacer todos los pasos anteriores hasta que se cumpla un criterio de parada.

2.6 Realineamiento de Viterbi

Después de la etapa de agrupamiento, es muy común la aplicación de técnicas para el realineamiento de los límites entre segmentos y así mejorar el rendimiento final del sistema. La técnica más utilizada es el realineamiento de Viterbi.

El proceso de realineamiento de viterbi puede verse de forma esquemática en la **Figura 9**:

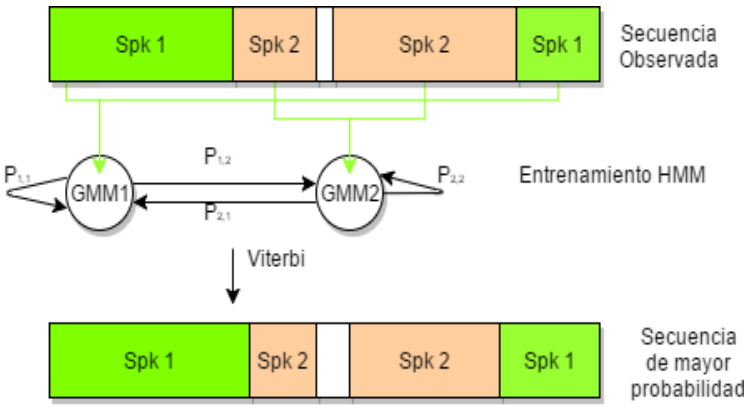


Figura 9. Esquema Realineamiento de Viterbi

Dada una secuencia de segmentos asociados a distintos *clusters*, el primer paso es el entrenamiento de un modelo oculto de Markov (HMM) con los *clusters* de salida de la etapa anterior, en el que cada estado oculto del HMM es representado por una GMM entrenada con los datos del clúster correspondiente. Cada estado oculto se puede modelar con un GMM de 8 gaussianas con una matriz de covarianza completa entrenadas con el algoritmo de *Expectation – Maximization (EM)*. Para leer más sobre HMM, GMM y EM ver (5) (6).

Dado el HMM, definido por las probabilidades de que un segmento sea de cada estado y por las probabilidades de transición entre cada uno de los estados, y dada la secuencia de segmentos, el algoritmo de Viterbi permite obtener la secuencia de modelos ocultos más probable, de forma que algunos vectores de características inicialmente asociados a un *cluster* pueden reasignarse al *cluster* contiguo.

2.7 Técnicas de Reconocimiento de locutor

Como ya hemos comentado en los puntos anteriores, en las etapas de segmentación, clustering y realineamiento de Viterbi es esencial el modelado de los conjuntos de vectores de características para distinguir correctamente los tramos correspondientes a distintos locutores. Este problema es análogo al que surge en las aplicaciones de reconocimiento de locutor, donde deben compararse dos grabaciones de voz (en las que solo aparece un único locutor) para decidir si se trata o no de la misma persona.

2.7.1 Modelos de Mezclas de Gaussianas (GMM)

El estado del arte en reconocimiento de locutor ha estado dominado durante muchos años por los modelos de mezclas de gaussianas (***Gaussian Mixture Models GMMs)***. Estos modelos representan la distribución de las características de un locutor mediante la suma ponderada (p_i) de componentes gaussianas, cada una descrita por un vector de medias (μ_i) y una matriz de covarianzas (Σ_i). La técnica más habitual para el entrenamiento de estos modelos es el algoritmo *Expectation -Maximization* (EM) (5).

$$p(\vec{x}|\lambda_s) = \sum_{i=1}^{M} p_i N(\vec{x}|\overrightarrow{\mu_i}, \Sigma_i) \quad (2.10)$$

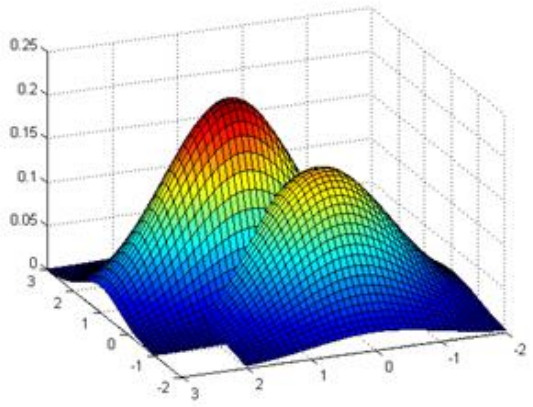


Figura 10. Ejemplo de GMM de dos componentes

En ocasiones, no se tiene más que un limitado conjunto de datos de entrenamiento para modelar correctamente un locutor específico mediante un GMM. Para solventar este problema surge el esquema ***GMM-UBM***, que consiste en la adaptación del modelo del locutor en cuestión a partir de un modelo universal (*Universal Background Mode, UBM*), que representa la distribución general de características en una población de locutores. La obtención del modelo adaptado para un locutor específico se realiza mediante *Maximum a Posteriori (MAP).*

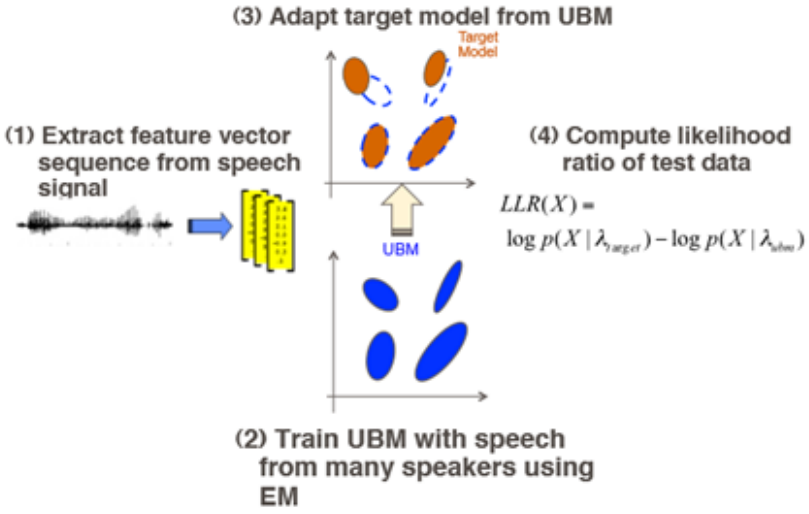


Figura 11. Entrenamiento de un sistema GMM-UBM (5)

Así pues, y como observamos en la **Figura 11,** un sistema GMM-UBM se basa en el entrenamiento de un UBM mediante audios de múltiples locutores. Posteriormente, para cada locutor específico a modelar, se adapta el UBM a las características del locutor. Finalmente, para un audio de test, se calcula el ratio de verosimilitudes de los vectores de características de éste (X) dado el modelo adaptado de locutor frente el dado el modelo UBM.

Basado en este esquema se han desarrollado técnicas que utilizan análisis factorial para representar al locutor en un espacio vectorial de dimensión reducida (comparada con el número de parámetros del modelo GMM) donde pueden aplicarse de forma eficiente técnicas de compensación de canal.

2.7.2 Modelos basados en Supervectores

Los ***"supervectores"*** (7) (8) son una representación de un modelo de locutor GMM en forma vectorial, de tal forma que el locutor puede identificarse con un punto en un espacio de coordenadas. En el contexto de un sistema GMM-UBM los distintos locutores tienen un origen de coordenadas común, las medias del UBM, y solo difieren en los valores de sus medias, pudiéndose representar por tanto de en forma vectorial mediante:

$$M = m + u \quad (2.11)$$

Siendo M el supervector del modelo de locutor adaptado, m es la concatenación de las medias del UBM y u es la variación debida a la identidad del locutor. El principal problema del uso de supervectores es que la comparativa entre los modelos de entrenamiento y los datos de test es de muy alta dimensión: CxD siendo C el número de componentes del UBM y D el número de características. Y, por lo tanto, supone un coste computacional muy alto.

2.7.3 Técnicas basadas en el análisis de factores: JFA e i-vectors

Para evitar esta complejidad computacional aparecen técnicas basadas en "análisis de factores" que representan las variaciones debidas al locutor y/o el canal en espacio de menor número de dimensiones. Las dos técnicas más utilizadas son ***JFA*** y los **i-vectors**, siendo estos últimos la técnica utilizada en este trabajo de fin de grado.
En ***Join Factor Analysis o JFA*** (9) se dividen las variaciones debidas al canal, al locutor y los componentes residuales como se muestra en la referencia (10).

$$M = m + Vy + Ux + Dz \text{ (2.12)}$$

Siendo en (2.12) *V, U y D* matrices dependientes del locutor, del canal y el componente residual respectivamente, que se entrenan y, a partir de las cuales, se obtienen los factores de variabilidad entre locutores *y*, canales *x*, y componentes residuales *D*. Para una descripción detallada del cálculo de estos (11).

Los i-vectors (10) representan de forma conjunta la variación debida al locutor y al canal en un mismo espacio de dimensión reducida, denominado espacio de variabilidad total. Así:

$$M = m + Tw \text{ (2.13)}$$

Donde *m* representa la concatenación de los vectores de medias de un UBM, *M* un modelo de locutor adaptado de igual dimensión, y por último el término *Tw* es una desviación que convierte el UBM en un modelo adaptado, y, por lo tanto, representa la información del locutor y del canal presentes en la grabación. En este último término T es la matriz de variabilidad total que transforma los supervectores de locutor (M) a un subespacio de menor número de dimensiones *w* denominados i-vector.

El "análisis factorial" calcula cuales son las dimensiones que más varían entre el modelo adaptado *M* y el modelo universal m, y estas son las que mantiene el i-vector, para el estudio del análisis factorial (12).

El modelado de locutores en forma vectorial mediante la utilización de i-vectors permite en la práctica una representación más discriminativa de los locutores, con menor coste computacional pues las dimensiones del i-vector son mucho menores y la utilización de medidas de distancia para la comparación de i vectors de diferentes segmentos sencillas. Un ejemplo de medida de distancia para determinar si dos segmentos modelados cada uno por un i vector provienen del mismo locutor o de uno diferente es la distancia coseno (8). Esta medida se basa en la evaluación del valor del coseno del ángulo que comprendido entre ambos vectores. Esta función proporciona un valor igual a 1 si el ángulo comprendido es 0 y por lo tanto si ambos vectores tienen la misma dirección y mismo sentido, y ambos segmentos formarían parte del mismo conjunto.

Basándonos en esta analogía mencionada anteriormente con el reconocimiento de locutores, este trabajo utilizará i-vectors en diferentes etapas del sistema, concretamente en la extracción de características, generando un flujo de i-vectors (Apartado 3.2.1) que esperamos que diferencie mejor los locutores que los coeficientes MFCC, y en la etapa de agrupamiento (Apartado 3.2.2), esperando que el modelado de los locutores mediante i-vectors tras la segmentación, permita diferenciarlos mejor.

3 Diseño

Tras repasar los diferentes aspectos teóricos que han sido estudiados a lo largo de este trabajo, este apartado ofrece una descripción del sistema de referencia inicial y de los dos sistemas basados en i-vectors desarrollados.

3.1 Sistema de referencia

El sistema de referencia, basado en el sistema LIUM de diarización de locutores y cuya descripción detallada se puede ver en (13) sigue la estructura típica de un sistema diarización, descrita en el Apartado 2.1 basada en las etapas de: Extracción de características, Speech Activity detection, segmentación, agrupamiento y realineamiento de Viterbi.

3.1.1 Extracción de características

El software utilizado para la extracción de características es Kaldi (14) . Este módulo extrae un vector de características MFCCs cada 10 ms mediante ventanas Hamming de 20 ms con un 50% de solapamiento (ver Apartado 2.2.1).

Para cada ventana se calcula un vector de 20 coeficientes MFCC mediante un banco de 25 filtros triangulares equiespaciados a lo largo de todo el espectro disponible (0-8.000 Hz).

3.1.2 Speech Activity Detector (SAD)

El detector de actividad de nuestro sistema de referencia se basa en el análisis de las trayectorias de los armónicos en un espectrograma logarítmico para cada uno de las tramas de audio, mencionado en el Apartado 2.3.3 de este trabajo.

El proceso realizado consiste primero en la normalización de la amplitud del audio en ventanas triangulares de 5 segundos de duración y un solapamiento del 50% con una ganancia variable inversamente proporcional al cuadrado de la energía dentro de la ventana.

Posteriormente se obtiene el espectrograma logarítmico. Para ello se realizan los siguientes pasos:

- Se divide la señal mediante ventanas Hamming de 30 ms cada 10 ms.
- Se calcula la FFT para cada una de las ventanas.
- Se divide la energía de cada ventana en un eje de frecuencia repartido en 6 octavas cada una de 40 columnas logarítmicas.

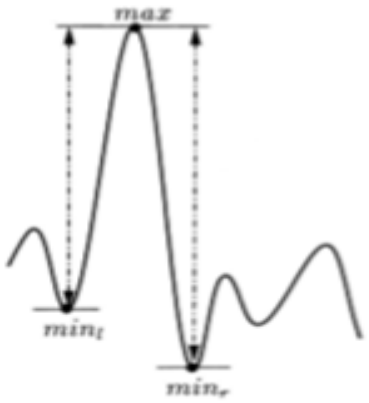
Para cada frame se calcula la matriz de correlación cruzada con un offset $\in$ [-4,4] y una desviación frecuencial l $\in$ [-10,10]. Y se calcula $R = r_{xcorr} - r$.

3.1.3 Segmentación

La etapa de segmentación en este sistema se divide en dos etapas:

La primera, la segmentación como tal, Apartado 2.4, se utilizan ventanas de 5 segundos con saltos de 100 ms Las características de cada una de las mitades de la ventana son modeladas mediante una GMM de 1024 componentes y con matriz covarianza completa.

La "distancia" entre ambas distribuciones es calculada mediante la medida GLR (Apartado 2.4.2). Para determinar si un máximo local de la curva de distancia corresponde a un cambio de locutor se realiza mediante la diferencia entre su valor y los valores de los mínimos adyacentes como se puede ver en la **Figura 12.** Detección de puntos de cambio en la etapa de segmentación y en **(**3.1**).**



$$|d(\max) - d(min)| > \alpha\sigma \quad (3.1)$$

Figura 12. Detección de puntos de cambio en la etapa de segmentación

Siendo σ la desviación típica de la medida y α un umbral seleccionado. Si se cumple (3.1**)** para ambos mínimos, entonces el máximo de la curva indica un punto de cambio de locutores. Para nuestro sistema α=0.5. Además, si la distancia entre dos máximos es menor que 1s entonces el de menor amplitud es descartado, para así conseguir una limitación de segmentos de corta duración y la aparición de falsos puntos de cambio.

Tras este proceso de segmentación tiene lugar una etapa de agrupamiento lineal como refinamiento ΔBIC de la segmentación. El objetivo de este agrupamiento es eliminar puntos de cambio espurios y así refinar los segmentos obtenidos del paso previo. Esta etapa consiste en, empezando por el primer punto de cambio, obtener la medida ΔBIC, descrita en el Apartado 2.4.1, para cada par de segmentos contiguos modelados cada uno como una gaussiana con una matriz de covarianza completa

El refinamiento fusiona segmentos consecutivos mediante los siguientes pasos:

Empezando con el primer segmento, se calcula la medida de distancia para cada par de segmentos adyacentes. Si ΔBIC < 0 dado un λ, inicialmente λ =4.8, el punto de cambio se descarta y ambos segmentos se fusionan en un nuevo segmento modelado con una nueva gaussiana. Posteriormente se realiza el mismo proceso para el nuevo segmento y su adyacente.

Este proceso se repite hasta que la distancia entre cada par de segmentos adyacentes ha sido calculada.

3.1.4 Agrupamiento Lineal y Agrupamiento AHC

Tras esto, se realiza un clustering jerárquico Aglomerativo AHC (ver Apartado 2.5.2) que, al igual que el paso anterior, utiliza ΔBIC como medida de distancia entre clusters no adyacentes modelados con una gaussiana con matriz de covarianza diagonal.

El proceso del clustering Aglomerativo es similar al clustering lineal de la etapa anterior. Si se cumple el criterio ΔBIC < 0 dado un λ, inicialmente λ =4.8, ambos *clusters* se fusionan. El nuevo segmento se modela con una nueva gaussiana y se calcula de nuevo la distancia entre el nuevo cluster y el resto. Si, por lo contrario ΔBIC > 0, los clusters no se fusionan y se analiza la distancia entre el siguiente cluster y el resto. El criterio de parada de este proceso es cuando la métrica entre cada par de *clusters* es mayor que 0.

3.1.5 Realineamiento de Viterbi

Finalmente, las posiciones de los puntos de cambio de locutor se refinan aplicando un realineamiento de viterbi anteriormente explicado en el Apartado 2.6. Cada *cluster* es modelado con un *Gaussian Mixture Model GMM* de 8 componentes. Se crea un *left-to-right HMM* (13) con todos los clusters representados por GMMs y se aplica el algoritmo de viterbi para obtener la secuencia de clusters más probable para dichas características.

3.2 Sistemas basados en i-vectors

Tras el estudio del sistema de referencia, y la adaptación de este a los datos de la evaluación, descrita en el Apartado 4.4.2, desarrollamos dos sistemas alternativos basados en la utilización de i-vectors en etapas específicas de la diarización esperando que tengan un mejor rendimiento que el sistema de referencia.

3.2.1 Sistema 1: Flujo de entrada de i-vectors

El primer sistema alternativo presentado consiste en la utilización de i-vectors en la etapa de extracción de características. Un extractor de i-vectors es un bloque de procesado basado en "análisis factorial" que nos permite representar un conjunto de vectores de características mediante un único vector que recoge la información discriminativa del locutor (y del canal) de forma muy eficiente, es por eso que, basándonos en el sistema presentado por ATVS a la evaluación Albayzín de 2010 (16), utilizamos un flujo de i-vectors de baja dimensión en vez de coeficientes MFCCs.

Cada i-vector se obtiene a partir de un conjunto de coeficientes MFCC mediante una ventana de 1,5 segundos que se desplaza cada 20 ms. Así pues, esta extracción de características de este sistema se basa en la obtención de un i-vector de dimensión 50 cada 20 ms.

La motivación es que, al ser estos i-vector de menores dimensiones, tal y como se comentó anteriormente en el Apartado 2.7.1, se espera que este flujo de i-vectors nos permita segmentar mejor que el flujo de vectores MFCC, pues en el reconocimiento de locutor se ha demostrado que, representar un conjunto de vectores mediante esta técnica, y aplicar una simple distancia coseno, permite diferenciar mejor entre locutores que el modelado directo de la distribución de MFCCs.

El resto de las etapas del sistema, el detector de actividad, la segmentación y el agrupamiento es igual que el sistema de referencia, pero teniendo como entrada un flujo de i-vectors de baja dimensión extraídos mediante una ventana deslizante en lugar de un flujo de vectores MFCC.

3.2.2 Sistema 2: Agrupamiento de i – vectors

El segundo sistema basado en la utilización de i-vectors tiene las etapas de extracción de características, detección de actividad y segmentación iguales que las del sistema inicial (Apartado 3.1.1, Apartado 3.1.2 y Apartado 3.1.3). La etapa de agrupamiento es una etapa similar al proceso de reconocimiento de locutor, se trata de comparar dos segmentos de voz y decidir en función de una medida de distancia si provienen del mismo locutor o no.

Mientras que el sistema de referencia realiza esta comparación directamente a partir de los coeficientes MFCC modelados con una gaussiana para cada segmento, el agrupamiento de este sistema alternativo se basa en la comparación de los i-vector que definen a cada segmento esperando que se obtengan mejores resultados que los anteriores.

El proceso de *clustering* de este sistema se divide en las siguientes etapas:

- Para cada segmento, supuestamente homogéneo, resultante de la etapa de segmentación se obtiene el i-vector que lo describe, mediante la utilización de una matriz T de transformación previamente comentada, y cuyo entrenamiento queda descrito en Apartado 4.2.4. El tamaño de los i-vectors es de 600x1 mientras que el de los supervectores que definen cada segmento son, como hemos visto en el Apartado 2.7.2: Número de Componentes GMM x Coeficientes MFCC = 1024x20 = 20840. Y, por lo tanto, la matriz T de transformación tiene unas dimensiones de 20840 x 600.

- Posteriormente se comparan los i-vectors de diferentes conjuntos mediante la medida de distancia coseno. Si la distancia coseno entre los dos vectores es superior a un umbral, inicialmente marcado 0.426, entonces ambos i-vectors pertenecen al mismo conjunto.

Se repite este proceso hasta que no se pueda fusionar ningún *cluster* más. Posteriormente, no se realiza ningún tipo de realineamiento.

4 Desarrollo

El desarrollo de este trabajo se puede dividir en dos etapas. La primera, realizada hasta el 14 de octubre de 2016, consiste en la adaptación de nuestro sistema de referencia a los datos proporcionados en la evaluación Albayzín 2016 (11) . El objetivo de la segunda etapa, realizada posteriormente, consiste en la utilización de técnicas alternativas a las anteriormente utilizadas en cada una de las fases de la diarización y, sobre todo, en el estudio y desarrollo de técnicas basadas en i-vectors.

Para comprender el desarrollo de este trabajo, primero repasemos las características del entorno en el que lo hemos realizado.

4.1 Entorno de desarrollo

4.1.1 Software utilizado

Para el desarrollo de este trabajo de fin de grado se han utilizado varias herramientas:

Para la extracción de características se ha utilizado la herramienta para reconocimiento de voz Kaldi (14).

El resto de los módulos del sistema de diarización de locutores de referencia, detección de actividad, segmentación, *clustering* y realineamiento de Viterbi, además de las técnicas nuevas desarrolladas han sido desarrolladas en Matlab (17).

Los experimentos han sido realizados en un servidor con dos microprocesadores Xeon Quad Core E5335 a 2.0GHz (permitiendo 8 hilos de ejecución) y 16 GB de RAM.

Por último, para la medida del rendimiento se ha usado la herramienta proporcionada por la evaluación Albayzín con este fin. Esta herramienta se trata de un script de Perl (18) desarrollado por el US National Institute of Standards and Technology (NIST) para la medida de rendimiento de sistemas de diarización, y ha sido utilizada en evaluaciones similares organizadas por dicho instituto (Rich Transcription evaluations)

4.1.2 Evaluación Albayzín 2016 de diarización de locutores

La evaluación Albayzin 2016 de diarización de locutores (1) consiste en la generación de etiquetas con indicación de los intervalos en los que intervienen distintos locutores para un fichero de audio de test dados. Para la evaluación se proporcionan datos de entrenamiento y de desarrollo del sistema.

Para esta evaluación fueron proporcionados datos de entrenamiento y de desarrollo el 15 de junio de 2016 y, posteriormente, datos de evaluación el 15 de septiembre de 2016. Para esta evaluación, se utilizarán dos bases de datos.

La base de datos de las noticias catalanas del canal de televisión 3/24 TV, forman un total de 87 horas de audio (85% con voz) y fueron proporcionados como conjunto de entrenamiento. Estas grabaciones han sido utilizadas para el entrenamiento de la matriz de

varianza total T para el extractor de i-vectors (UBM y matriz T) del sistema basado en el agrupamiento de i-vectors.

La base de datos donada por la Corporación Aragonesa de Radio y Televisión (CARTV) que conforman un total de 20 horas (92% con voz) de audio para el desarrollo (5 horas repartidas en 32 audios de longitud variable) y la evaluación final del sistema (15 horas en 73 grabaciones). Las grabaciones de desarrollo han sido utilizadas para la adaptación del sistema, mediante el barrido de los diferentes parámetros característicos de nuestros sistemas. Los resultados descritos en el Apartado 5 se obtuvieron sobre este conjunto de desarrollo. Las grabaciones de evaluación sólo se emplearon para generar los resultados enviados a la evaluación.

Todo el audio proporcionado ha sido suministrado en formato PCM, Monocanal y de 16 kHz de frecuencia de muestreo.

Los resultados obtenidos tenían que ser entregados antes del 15 de octubre junto con una descripción de los sistemas presentados a la evaluación y, hasta el 31 de ese mismo mes no se publicaban los resultados a los participantes.

4.1.3 Formatos de Salida: RTTM

La salida habitual de un sistema de diarización consiste en un fichero de texto en el cual aparecen las etiquetas que indican cada una de las intervenciones de los diferentes locutores indicando el intervalo en el que se dan lugar, y un identificador de locutor. Estas etiquetas además pueden incluir información adicional de utilidad para sistemas posteriores, como por ejemplo el género del locutor.

Pese a que todos los formatos que aparecen en la literatura son similares, ya que todos describen la misma información, destacamos el formato RTTM pues es el usado en la evaluación Albayzín y que a su vez viene de evaluaciones organizadas anteriormente por el NIST denominadas Rich Transcription evaluations (19).

El formato RTTM o “Rich Transcriprion Time Marked” (20) es el formato de referencia para los sistemas de procesado de audio para la extracción de metadatos.

Un ejemplo de salida RTTM sería el siguiente:

```
SPEAKER session2 1   0.000   4.358   <NA>   <NA>   spk1   <NA>
SPEAKER session2 1   6.487   12.322  <NA>   <NA>   spk2   <NA>
SPEAKER session2 1   13.025  17.524  <NA>   <NA>   spk1   <NA>
SPEAKER session2 1   18.221  21.030  <NA>   <NA>   spk3   <NA>
```

El primer campo de la salida RTTM indica la fuente de audio, y en este caso la presencia de un locutor.
El segundo campo corresponde el nombre de la sesión de audio que está siendo evaluada.
El tercero es el número de canales del fichero de audio.

El cuarto campo indica el momento de inicio de la intervención y el quinto es la duración de dicha intervención ambos en segundos.
Todos los campos marcados como <NA> son campos no obligatorios reservados para información adicional de utilidad en otros sistemas, ya que este es un formato de propósito general para distintas aplicaciones.

Por último, el octavo campo corresponde a la etiqueta de cada locutor. La única condición que tiene que cumplir esta etiqueta es que tiene que ser la misma para intervenciones del mismo locutor y diferente para las demás.

4.1.4 Medidas del Rendimiento: DER

Para comprobar el funcionamiento de nuestro sistema es necesario comparar su salida con las etiquetas reales (*ground truth*) del audio. La métrica más utilizada para la medida del rendimiento de un sistema de diarización de locutores es la tasa de error de diarización o DER (Diarization Error Rate)

La tasa de error de diarización (diarización) se puede definir, dado un audio de entrada, como el tiempo total incorrectamente etiquetado dividido por la duración total del audio hablado. De esta forma, debido a la ponderación temporal, los segmentos más largos tendrán mayor impacto sobre el rendimiento final del sistema que los segmentos cortos.

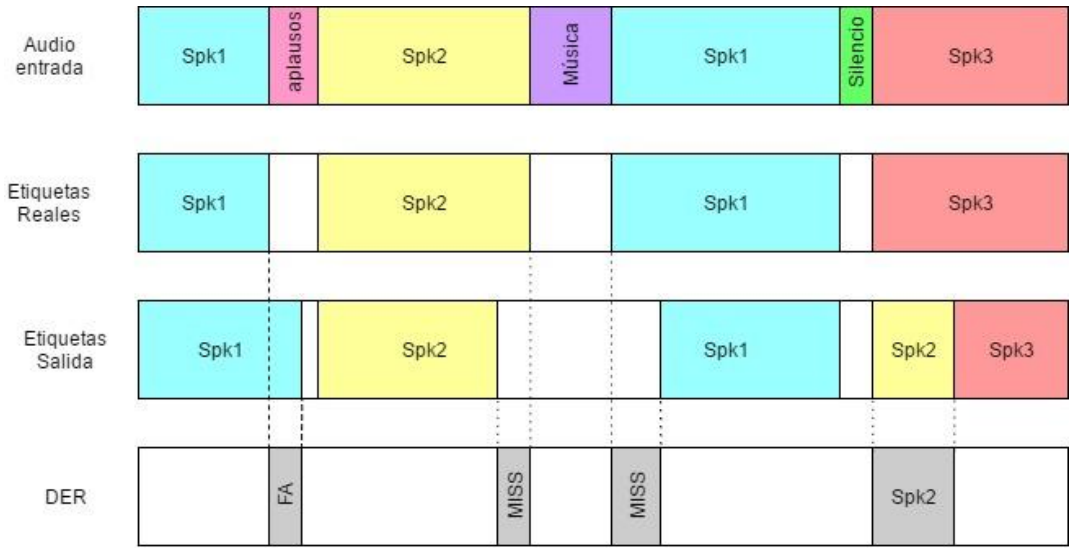


Figura 13. Componentes de la medida *Diarization Error Rate (DER)*

Como podemos ver en la **Figura 13** el error total de diarización se puede dividir en 3 tipos de error al etiquetar incorrectamente el audio:

- **False Alarm (FA)** es la fracción del tiempo etiquetado como habla mientras no lo es. Es decir, es el tiempo de silencio, o por ejemplo música, etiquetado incorrectamente como voz.

- **Missed Speech (MISS)** es, al contrario que el FA, la fracción de tiempo con voz que no ha sido etiquetada como tal. Estos dos tipos de error dependen exclusivamente del rendimiento del detector de actividad independientemente del proceso en sí de diarización.

- **Speaker Error (SPKE)** es la fracción del tiempo de habla total asignado a un hablante incorrecto. Este error depende exclusivamente del proceso de segmentación y clustering y no del detector de actividad. Este error es el más significativo de los tres que forman la tasa de error de diarización.

4.2 Evolución

Una vez que conocemos el entorno de desarrollo en el que ha tenido lugar este trabajo, veamos la evolución que ha tenido el trabajo.

4.2.1 Adaptación a la evaluación

En la primera etapa del desarrollo de este trabajo de fin de grado se realizan modificaciones en los parámetros que describen nuestro sistema de referencia para conseguir un mejor rendimiento con los datos proporcionados.

El listado de parámetros a modificar es grande y el conjunto de posibles combinaciones de valores de entrenamiento del sistema es aún mayor. Así pues, para la adaptación de nuestro sistema de referencia a los datos de la evaluación Albayzín hemos realizado una variación (*tunning* o barrido) de aquellos parámetros con mayor impacto en el rendimiento final hasta conseguir los mejores resultados.

Podemos agrupar las pruebas realizadas en función de la etapa de origen del parámetro que hacemos variar. De esta forma, algunos de los parámetros sobre los que podemos realizar un *tunning*:

En la detección de actividad:

- Se modifica el umbral de decisión a partir del cual es clasificado un segmento de audio como voz o no.

En la etapa de segmentación se modifican los siguientes valores:

- α : Factor que se le aplica a la desviación típica, como vemos en el Apartado 3.1.3, de la medida de distancia para determinar qué máximos de dicha curva corresponden a cambios de locutor.

- *d:* Separación máxima entre dos máximos de la curva de distancia. Se han probado 100 ms y 50 ms.

- λ : Factor que determina el umbral de decisión del algoritmo ΔBIC como se ve en la fórmula (2.7) de la etapa de refinamiento de la segmentación.

- Utilización de una matriz de covarianzas completa o diagonal para el modelado de las características en cada ventana.

En la etapa de clustering:

- λ : Factor que determina el umbral de decisión del algoritmo ΔBIC, como se ve en el Apartado 2.4.1. Tanto para el refinamiento de la segmentación (*clustering* lineal) como para el *custering* AHC han sido probados valores entre 1,5 y 6,5.

- Utilización de una matriz de covarianzas completa o diagonal para el modelado de cada *cluster*.

En el realineamiento de Viterbi:

- Se han realizado pruebas con y sin realineamiento de Viterbi.

Los resultados de todas estas modificaciones quedan reflejados en el Apartado 5.1

4.2.2 Incorporación de coeficientes derivativos.

Tras la realización de las pruebas anteriores, y con el objetivo de aumentar el rendimiento de nuestro sistema de diarización, observamos cómo se comporta el sistema al modificar su entrada y, en vez de realizar la diarización a partir de 20 coeficientes estáticos MFCCs, se realizará la diarización incorporando coeficientes derivativos (ver Apartado 2.2.3), dando lugar a vectores de dimensión 20 x 2 = 40 MFCCs.

4.2.3 Sistema 1: Flujo de entrada de i-vectors

Una vez obtenidos los mejores resultados con el sistema de referencia, comenzamos el desarrollo del sistema alternativo basado en la extracción de i-vectors como características sobre las cuales se realizará la segmentación (ver Apartado 3.2.1). El desarrollo de este sistema se basa la adaptación de las otras etapas del sistema de diarización de referencia (segmentación y agrupamiento) a la nueva extracción de características y el posterior barrido de los parámetros anteriores.

Los resultados obtenidos se encuentran resumidos en el Apartado 5.2.

4.2.4 Sistema 2: Agrupamiento de i-vectors

Por último, y con el objetivo de profundizar en el estudio de los i vectors e intentar mejorar los resultados obtenidos anteriormente. Se comienza el entrenamiento del sistema descrito en el Apartado 3.2.2 Este sistema consistirá en las etapas de extracción de características, detección de actividad y segmentación ya desarrolladas en el sistema de referencia. La novedad de este sistema es la etapa de agrupamiento o *clustering.*

Para el entrenamiento de esta etapa de *clustering* seguimos las siguientes etapas:

Primero realizamos el entrenamiento del modelo universal UBM con los ficheros de entrenamiento, con una duración total de 87 horas. Y modelamos todos los ficheros de entrada mediante un GMM de 1024 gaussianas con matrices de covarianza diagonales y medias *m.*

Posteriormente, y desarrollando la ecuación (2.13) tal que:

$$M - m = Tw \quad (4.1)$$

Tenemos que, para obtener la matriz T de transformación de supervectores a i vectors, hay que obtener lo que se denominan los estadísticos propios de cada locutor que representan la desviación de cada locutor frente a las medias del UBM.

Para obtener los estadísticos se modela cada locutor obtenido de la base de datos de desarrollo, mediante un GMM de iguales características que el UBM y se obtiene su desviación frente al UBM previamente modelado.

Una vez tenemos estos estadísticos podemos entrenar la matriz de transformación 'T'. Y, una vez entrenada la matriz de variabilidad completa, para cada segmento de audio de entrada podemos calcular su i vector asociado.

Tras el entrenamiento de la matriz T, modificamos el sistema de referencia para que, tras la etapa de segmentación, cuyo resultado son los segmentos de voz en los que en un principio se han detectado cambios de locutor, se calculen los i vectors de cada segmento de salida y se realice el *clustering* sobre estos.

La agrupación de segmentos de audio, definidos cada uno por i-vectors diferentes, se realiza mediante un clustering jerárquico Aglomerativo (*AHC).* Para definir los parámetros iniciales que definirán el rendimiento final de nuestro sistema realizamos un estudio de la impureza de *cluster* y la impureza de clases.

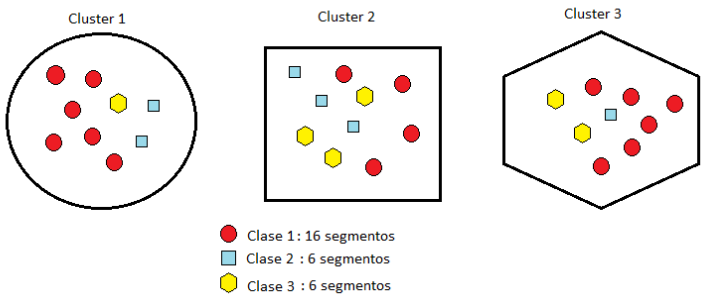


Figura 14. Impureza de *Cluster* – Impureza de Clase

La **Figura 14** nos permite entender de forma gráfica el significado de los conceptos de impureza de clase y de *cluster.* Definimos la impureza de cluster (21) como el porcentaje de segmentos dentro de ese *cluster,* que en realidad no forman parte de la clase asociada a este. Para el cluster 1 de la figura la impureza de *cluster* sería:

$$I_{cluster1} = \frac{n_{clase2|cluster1} + n_{clase3|cluster1}}{n_{cluster1}} = \frac{1+2}{9} = 0{,}3 \rightarrow 30\%$$

Por otro lado, denominamos impureza de clase al porcentaje de segmentos pertenecientes a una clase que no están en su *cluster* asociado. Siguiendo el ejemplo anterior, para la clase 1 tenemos:

$$I_{clase1} = \frac{n_{clase1|cluster2} + n_{clase1|cluster3}}{n_{clase1}} = \frac{3+6}{16} = 0{,}5625 \rightarrow 56{,}25\%$$

Así pues, los parámetros que definirán la etapa de agrupamiento de los i vectors serán escogidos inicialmente en función de las impurezas medias de *cluster* y las impurezas medias de clase de cada una de las grabaciones de desarrollo, y posteriormente en función del rendimiento final del sistema como se puede observar en el Apartado 5.3

Los parámetros del clustering son:

- El umbral o *cutoff* de decisión de la medida de distancia seleccionada.

- La distancia considerada a la hora de comparar dos clusters o *linkage*. Las opciones más habituales son:

  "*Single*": La medida de distancia entre dos clusters es la distancia para los vectores más cercanos (uno de cada *cluster*).

  "*Average":* La distancia entre dos *clusters* es el promedio de distancias para cada par de vectores par de vectores.

  "Complete": La distancia entre dos *clusters* es la distancia para los vectores más lejanos (uno de cada *cluster*).

- Medida de distancia: La medida de distancia que se utiliza para definir si dos segmentos definidos por un i vector forman parte del mismo cluster es la medida coseno. Dados dos segmentos, al estar definidos de forma vectorial cada uno de ellos, coseno entre ambos es cercano a cero, significa que ambos tienen la misma orientación espacial, y, por lo tanto, pertenecen al mismo cluster.

Por último, se realiza un barrido de los parámetros de la segmentación definidos en el Apartado 4.4.2

5 Integración, pruebas y resultados

Tras describir el desarrollo de este trabajo, en este apartado se reflejarán los resultados de cada uno de los procesos llevados a cabo. Podemos dividir las pruebas realizadas en los siguientes puntos:

5.1 Adaptación del sistema de referencia a la evaluación Albayzín

En este apartado veremos el impacto que tiene sobre el rendimiento del sistema de referencia la alteración de los parámetros definidos anteriormente para cada una de las etapas de la diarización:

5.1.1 SAD

Lo primero que realizamos es una modificación del umbral del *Speech Activity Detector*, obteniendo los siguientes resultados:

Umbral Score	DER (%)	FA (%)	MISS (%)	SPKE (%)
50	46,96	1,60	24,40	20,90
35-inicial	38,66	3,50	9,80	25,40
20	**33,10**	**6,50**	**3,70**	**22,80**
15	34,98	7,70	3,70	23,60

Tabla 1. Variación del umbral del SAD

Como podemos observar, la variación del umbral del *Speech Activity Detector* es la única etapa de la diarización que afectará directamente a los errores de FA y MISS (definidos en el Apartado 4.1.4), y, por lo tanto, indirectamente al error de locutor.

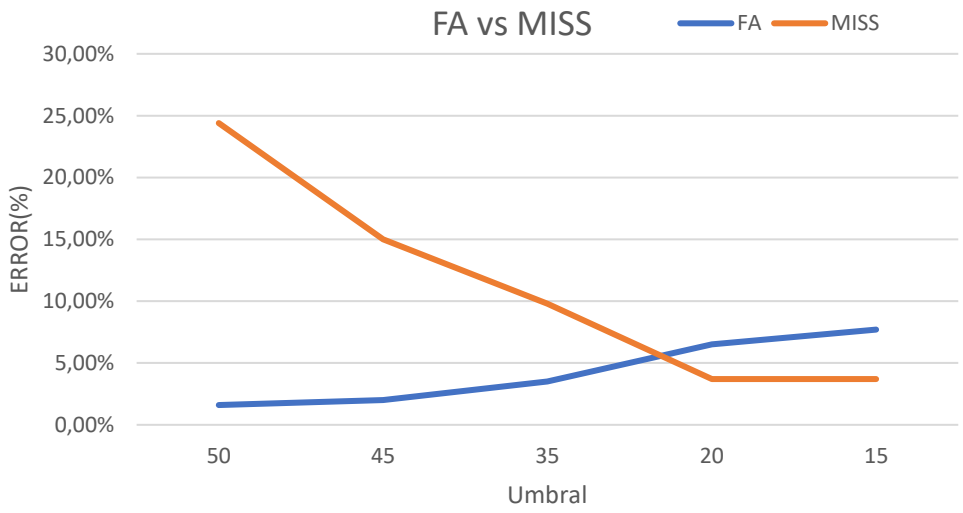


Figura 15. Selección del umbral del Speech Activity Detector

Para mejorar el rendimiento del sistema tenemos que encontrar un valor que estabilice ambos errores, idealmente el punto de cruce de las curvas representadas en la **Figura 15.** Selección del umbral del Speech Activity Detector. Sin embargo, como podemos ver en la **Tabla 1**, el umbral con el que se obtiene el mejor rendimiento (menor DER) es 20. Obteniendo un *33,10%* de DER, *5%* menos que la configuración de parámetros iniciales. Esta salida del SAD se utiliza para los 3 sistemas.

Además, podemos observar en la **Tabla 1,** para el mejor resultado obtenido, el componente del DER con mayor impacto en el rendimiento final es el error de locutor (SPKE). Para disminuir este error realizamos el barrido de los parámetros de la segmentación y del agrupamiento.

Con el valor del umbral seleccionado, comenzamos el barrido de los parámetros propios de las etapas de segmentación y del *clustering.*

5.1.2 Segmentación

En las siguientes tablas quedan registrados algunos de los resultados obtenidos tras el barrido de los diferentes parámetros más influyentes en el rendimiento del sistema de diarización:

- Segmentación: Refinamiento ΔBIC λ

λ	**DER (%)**	**FA (%)**	**MISS (%)**	**SPKE (%)**
6	44,02	6,50	3,70	33,70
4.8	*33,10*	*6,50*	*3,70*	*22,80*
3	**32,01**	**6,50**	**3,70**	**21,07**
2	38,57	6,30	3,80	28,40

Tabla 2. Sistema de Referencia. Variación del error en función del clustering lineal: λ

- Segmentación: Refinamiento ΔBIC: Utilización de matriz de covarianzas diagonal o completa

Σ	**DER (%)**	**FA (%)**	**MISS (%)**	**SPKE (%)**
Diagonal	43,56	6,50	3,70	33,30
Completa	**32,01**	**6,50**	**3,70**	**21,07**

Tabla 3. Sistema de Referencia: Variación del error en función del clustering lineal: Σ

- Segmentación: Tamaño de ventana deslizante.

$l\ (ms)$	**DER (%)**	**FA (%)**	**MISS (%)**	**SPKE (%)**
6.000	41,32	6,50	3,70	31,10
5.000	**32,01**	**6,50**	**3,70**	**21,07**
4.000	**38,71**	**6,50**	**3,70**	**28,50**

Tabla 4. Sistema de Referencia. Segmentación: Variación del error en función del tamaño de ventana

Tras el barrido de estos parámetros, podemos observar en la **Tabla 2** que conseguimos obtener un resultado menor al anterior para $\lambda = 3$, matriz de covarianza completa y tamaño de ventana deslizante igual a 5 segundos.

5.1.3 Clustering AHC

Con el objetivo de disminuir el error de etiquetado de locutor, modificamos los parámetros de la etapa de agrupamiento, que es determinante ya que es la etapa que genera la unión de clusters no adyacentes, y con esto, el etiquetado final de cada segmento.

Los parámetros modificados de esta etapa son:

- *Clustering* AHC: λ

λ	**DER (%)**	**FA (%)**	**MISS (%)**	**SPKE (%)**
6	29,34	6,3	3,50	19,5
5	***27,12***	***6,3***	***3,20***	***17,6***
4,8	**32,01**	**6,50**	**3,70**	**21,07**
3	36,84	6,6	3,70	26,6

Tabla 5. Sistema de Referencia: Variación del error en función del clustering AHC: λ

- *Clustering* AHC: Utilización de matriz de covarianzas diagonal o completa:

Σ	**DER (%)**	**FA (%)**	**MISS (%)**	**SPKE (%)**
Completa	29,38	6,50	3,70	19,1
Diagonal	***27,12***	***6,30***	***3,20***	***17,6***

Tabla 6. Sistema de Referencia: Variación del error en función del clustering AHC: Σ

Tras la modificación de los parámetros del clustering AHC conseguimos disminuir el DER un 5% con respecto a las pruebas anteriores.

5.1.4 Realineamiento de Viterbi

Una vez modificados estos parámetros para obtener el mejor rendimiento del sistema para los datos de desarrollo. Probamos el sistema con el módulo de realineamiento de Viterbi obteniendo y el uso de coeficientes derivativos:

$Viterbi$	**DER (%)**	**FA (%)**	**MISS (%)**	**SPKE (%)**	**Tiempo(s) – audio 15 min**
Si	**26,95**	**6,30**	**3,30**	**17,4**	**54,25s**
No	*27,12*	*6,30*	*3,20*	*17,6*	*51,00s*

Tabla 7.Sistema de Referencia. Resultados con Realineamiento de viterbi

$Coef.\Delta$	DER	FA	MISS	SPKE	Tiempo(s) – audio 15 min
Si	36,16	6,60 %	3,70 %	25,90%	55,60
No	**26,95**	**6,30 %**	**3,30 %**	**18,40 %**	**54,25s**

Tabla 8.Sistema de Referencia. Resultados con coeficientes derivativos.

Del análisis de los resultados del sistema de referencia podemos extraer varias conclusiones:

Primero, el error de locutor, es el que mayor impacto tiene en nuestro sistema tomando un rango de valores de 17,4% (**Tabla** 7) hasta 33,70% (**Tabla 2**, ***λ =6***)

El Realineamiento de Viterbi (**Tabla7**), mejora muy poco el rendimiento final del sistema y le añade carga computacional. Para la evaluación de los siguientes sistemas no realizaremos pruebas con el módulo de realineamiento de viterbi.

El fracaso en el uso de coeficientes derivativos (**Tabla 8**) se debe a que el sistema no está adaptado para coeficientes de dimensiones tan grandes (40). Tenemos el doble de dimensiones, pero seguimos teniendo los mismos datos para modelar las características pues el tamaño de la ventana deslizante es el mismo. Para la evaluación de los siguientes sistemas no realizaremos pruebas con coeficientes derivativos.

Los valores que más afectan al rendimiento del sistema son los dos factores de penalización de cada una de las etapas de segmentación y agrupamiento (**Tabla 2** y **Tabla 5**).

Tras el barrido de los parámetros con mayor impacto sobre el rendimiento final del sistema, los mejores resultados obtenidos para los datos de desarrollo son un 26,95% y un 27,12% de tasa de error de diarización, con y sin realineamiento de viterbi respectivamente.

Los tiempos de ejecución de las diferentes etapas de la diarización vienen definidos en la **Tabla 9** para un audio de 15 minutos.

Etapa	Tiempo (minutos)/audio 15 mins
Extracción de características	**0,25**
Detección de Actividad	**5,2**
Diarización de locutores	**0,85**

Tabla 9. Sistema de Referencia. Tiempos de ejecución

Podemos ver en la tabla anterior que, para el sistema de referencia, el proceso total llevado a cabo por el sistema de diarización es mucho más rápido que el tiempo real, pues para cada audio de 15 minutos invierte un total de 6,3 minutos.

5.2 Sistema alternativo: Flujo de entrada de I-vectors.

Después de adaptar el sistema de referencia a los datos de desarrollo de la evaluación, se desarrolló el sistema basado en la utilización de un flujo de i-vectors en vez de coeficientes MFCC como características de entrada al sistema de segmentación.

Mientras que en la etapa anterior se realizó un barrido siguiendo paso a paso las etapas de la diarización modificando los parámetros iniciales, definidos a lo largo del Apartado 3, para este sistema se realizó un barrido primero de los parámetros que, tras los resultados anteriores, sabemos que tienen mayor impacto sobre la DER final del sistema.

Así pues, podemos dividir las pruebas en función del parámetro modificado de la siguiente forma:

5.2.1 Variación de *λ* para el agrupamiento lineal y agrupamiento AHC

En esta etapa modificamos el coeficiente de penalización o *λ*, para la segmentación (etapa intermedia de *clustering* lineal) y para el agrupamiento AHC obteniendo los siguientes resultados:

- Agrupamiento lineal: *λ*

λ	**DER (%)**	**FA (%)**	**MISS (%)**	**SPKE (%)**
6	**64,02**	**6,50**	**4,20**	**53,30**
4,8	66,72	6,50	4,10	56,10
3	67,03	6,50	4,20	56,30
2	69,12	6,50	4,20	58,40

Tabla 10. Sistema de flujo de entrada de i-vectors. *Clustering* lineal: *λ*

Con los primeros resultados del barrido de parámetros, ya vemos que el error de este sistema va a ser considerablemente mayor al anterior. Si comparamos los resultados iniciales del sistema, este sistema parte de un error inicial de *66,72%* frente al 33,10% del anterior.

- Agrupamiento AHC: *λ*

λ	**DER (%)**	**FA (%)**	**MISS (%)**	**SPKE (%)**
4,8	64,05	6,50	4,20	53,30%
3	45,32	6,50	4,00	34,70%
2	**38,49**	**6,50**	**4,00**	**27,49%**
1,5	70,18	6,50	4,20	59,50%

Tabla 11. Sistema de flujo de entrada de i-vectors. Clustering AHC: *λ*

Como ya introdujimos anteriormente, la etapa de agrupamiento es aquella que mayor impacto tiene sobre la DER final para los resultados obtenidos. Esto se puede reflejar en la Tabla 11 pues conseguimos disminuir el error final hasta *38,49%* (diferencia de *28%* con respecto al resultado anterior).

5.2.1 Variación de otros parámetros de diarización

Tras los resultados obtenido en Apartado 5.2.1, realizamos la variación de los parámetros que faltan para completar el barrido, realizado en Apartado 5.1.2 y Apartado 5.1.3:

- Clustering Lineal: Utilización de matriz de covarianzas diagonal o completa:

Σ	**DER (%)**	**FA (%)**	**MISS (%)**	**SPKE (%)**
Diagonal	44,32%	6,50%	4,00%	33,90%
Completa	**38,49%**	**6,50%**	**4,00%**	**27,90%**

Tabla 12. Sistema de flujo de entrada de i-vectors. Clustering Lineal: *Σ*

- Segmentación. Tamaño de ventana deslizante:

l (ms)	**DER (%)**	**FA (%)**	**MISS (%)**	**SPKE (%)**
6.000	40,72	6,50	4,00	30,20
5.000	**38,49**	**6,50**	**4,00**	**27,49**
4.000	44,62	6,50	4,00	34,10

Tabla 13. Sistema de flujo de entrada de i-vectors. Segmentación: Variación del error en función del tamaño de ventana

- Clustering AHC: Utilización de matriz de covarianzas diagonal o completa:

Σ	**DER (%)**	**FA (%)**	**MISS (%)**	**SPKE (%)**
Diagonal	**38,49**	**6,50**	**4,00**	**27,49**
Completa	89,46	6,50	4,00	78,9

Tabla 14. Sistema de flujo de entrada de i-vectors. Clustering AHC: Σ

De este análisis se pueden extraer varias conclusiones:

- Primero, como vemos en la **Tabla 12, Tabla 13** y **Tabla 14** modificando los parámetros iniciales (tamaño de ventana de 5 segundos, para el clustering lineal de la segmentación matriz de covarianzas completa, y para el clustering AHC matriz de covarianzas diagonal), no se consigue mejoría alguna, solo se ha consigue mediante el *tunning* del factor de penalización de ΔBIC.

- Por otra parte, el mejor rendimiento obtenido con este sistema es claramente peor al sistema de referencia (38,*49%* de DER frente a 26,*95*%).

- Esta alta tasa de error de diarización se debe a la extracción de i-vectors. En vez de entrenar nuestro extractor de i-vectors (UBM y matriz T) para los datos de desarrollo de esta evaluación, como se hace para el sistema de agrupamiento de i-vectors, utilizamos una matriz de transformación previamente entrenada sobre el conjunto de datos de train que se compone de grabaciones de audio en catalán.

- Por último, viendo el rendimiento del sistema en cuanto a tiempos de ejecución en la **Tabla 15**, y comparándolas con la **Tabla 9**, observamos que la etapa de extracción de i-vectors implica un gran coste computacional (8 veces la duración del audio) con respecto a la extracción de características MFFC, pues requiere la realización de operaciones con supervectores de muy altas dimensiones (ver Apartado 4.2.4).

  El tiempo de ejecución en la etapa de diarización como tal (segmentación y agrupamiento) disminuye gracias a la reducida dimensión de los i-vectors.

Etapa	Tiempo (min)/ audio 15 min
Extracción de i-vectors	**120**
Detección de Actividad	**5,2**
Diarización de locutores	**0,71**

Tabla 15. Sistema de flujo de entrada de i-vectors. Tiempos de ejecución

5.3 Sistema alternativo: Agrupamiento de I-vectors.

Como ha sido explicado en el Apartado 3.2.2, este sistema se basa en el modelado de cada segmento, determinado por la etapa de segmentación, por medio de un i-vector. Para decidir si dos segmentos pertenecen al mismo *cluster* se calcula la distancia coseno entre los dos i-vectors correspondientes.

Los parámetros sobre los cuales se puede realizar un barrido para este sistema son: el umbral de decisión, a partir del cual se decide si dos segmentos pertenecen al mismo *cluster* en función de la distancia coseno, y la distancia considerada a la hora de comparar dos *clusters* definida en el Apartado 4.2.4 y cuyos valores probados han sido las distancias *"average", "Single"* y *"Complete"*, definidas también en dicho apartado.

Para la elección de parámetros realizamos un estudio de la evolución media de la impureza de cluster y de la impureza de clase, en función de ambos parámetros. Obteniendo:

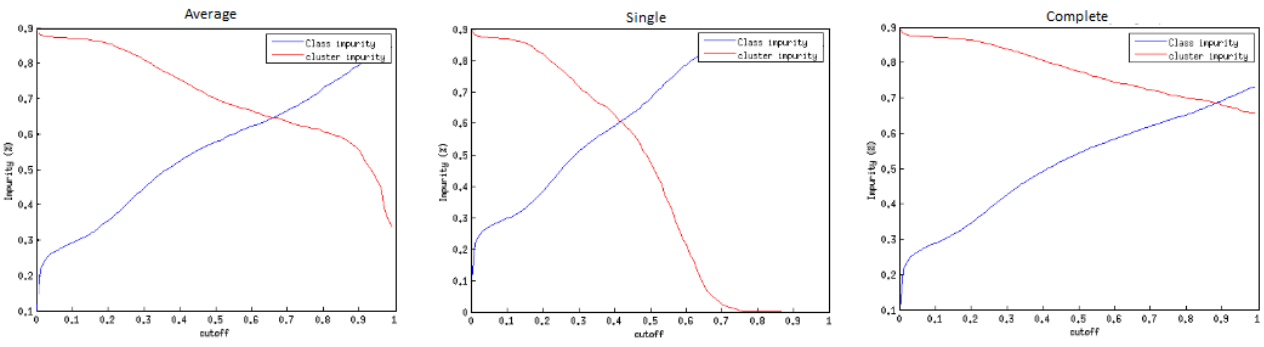


Figura 16. Estudio de la impureza de clase para el *clustering* de i – vectors.

Como podemos ver en la **Figura 16**, la distancia con la que, en principio, se van a obtener menor grado de impureza en el punto de equilibrio entre la impureza de clase y de *cluster* es en el caso del single para un umbral de 0,4 aproximadamente.

Por lo tanto, realizaremos un barrido cercano a dicho valor de umbral y para las configuraciones *single* y *average*, sabiendo de antemano que los resultados entre ambas distancias serán cercanos, aunque para la primera será un algo mejor:

Cutoff	DER	FA	MISS	SPKE	Tiempo(s)
0,20	30,08%	6,50%	4,80%	18,80	62,50s
0,25	28,45%	*6,50%*	*4,80%*	*17,20%*	*58,5s*
0,50	33,07%	6,50%	4,80%	21,80%	62,15s

Tabla 16. Sistema de Agrupamiento de i-vectors. Distancia average.

Cutoff	DER	FA	MISS	SPKE	Tiempo(s)
0,5	43,18%	6,50%	3,70%	32,9%	66s
0,42	37,52%	*6,50%*	*3,60%*	*27,2%*	*62,45s*
0,3	29,54%	6,50%	3,70%	19,30%	62,45s
0,25	**25,29%**	**6,40%**	**3,70%**	**15,00%**	**64,50s**
0,2	**27,11%**	**6,50%**	**3,70%**	**16,80%**	**66s**

Tabla 17. Sistema de Agrupamiento de i-vectors. Distancia single

De este análisis podemos extraer las siguientes conclusiones:

- Primero tal y como se observa en la **Tabla 16** y en la **Tabla 17,** la configuración que obtiene mejores resultados obtenidos es la distancia entre los segmentos más cercanos de cada uno de los *clusters* (*Single)* para determinar si éstos se unen o no.

- Por otra parte, observamos que el error de diarización disminuye con respecto a los dos sistemas anteriores (*25,9*% frente a 38,*45%* del segundo sistema y a 26,*95* del sistema de referencia), y, por lo tanto, podemos comprobar que la utilización de i-vectors en la etapa de agrupamiento aporta mejores resultados que la diarización basada en flujo de i-vectors.

- Además, y viendo la **Tabla 18**, comprobamos que el tiempo de ejecución de la diarización de locutores (segmentación y agrupamiento) aumenta con respecto a los Sistemas anteriores, pues en la etapa de agrupamiento se realiza el cálculo de los i-vector que supone un alto coste computacional debido a la alta dimensión de los supervectores a partir de los cuales se calculan.

Etapa	Tiempo (minutos)
Extracción de características	**0,25**
Detección de Actividad	**5,2**
Diarización de locutores	**1,05**
Cálculo de i-vectors	**0,30**

Tabla 18. Sistema de Agrupamiento de i-vectors. Tiempos de ejecución.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Este trabajo de fin de grado ha analizado diferentes técnicas de diarización mediante el estudio y/o desarrollo de 3 sistemas diferentes y la realización de pruebas en el contexto de la evaluación Albayzín 2016 de diarización de locutores.

Para el sistema de referencia, implementado por Cristian Sánchez Rodríguez y descrito en (2), las pruebas se han limitado a un proceso de adaptación de este sistema a los datos de la evaluación. Con una simple adaptación de parámetros hemos conseguido reducir la DER final de *44*% a 26,95%. Además, nos ha permitido el estudio de módulos de Realineamiento de Viterbi y extracción de características con coeficientes derivativos, aunque este último no haya resultado en una mejora del rendimiento.

Además de este sistema, este trabajo de fin de grado presenta un estudio de varias aproximaciones basadas en el modelado de locutores mediante i-vectors, estado del arte de los sistemas de reconocimiento de locutores actuales, y su aplicación en diferentes etapas de la diarización, en concreto, la extracción de características y el agrupamiento.

El primer sistema alternativo, también presentado a la evaluación, basado en la utilización de un flujo de i-vectors como entrada a la segmentación, no ha mejorado los resultados del sistema de referencia, teniendo como mejor DER final *38,49 %* , presumiblemente debido a la utilización de un extractor de i-vectors entrenado sobre un conjunto de datos en un idioma distinto al de test.

Por último, el desarrollo del sistema de diarización con agrupamiento basado en i-vectors, y el entrenamiento del UBM y matriz T necesarios para la extracción de estos, ha permitido un mejor estudio de los i-vectors además de ofrecer mejores resultados que los sistemas anteriores, registrando su mejor rendimiento en un *25*,29% de DER

6.2 Trabajo futuro

Este trabajo de fin de grado permite un gran abanico de posibles trabajos futuros:

De forma más inmediata, el entrenamiento del extractor de características con datos más parecidos a los de desarrollo de esta evaluación, y el uso de PLDA (22) (*Probabilistic Linear Discriminant Analysis)* como medida de distancia alternativa a la distancia coseno en el sistema basado en el agrupamiento de i-vectors, serían dos pruebas que enriquecerían el análisis realizado y posiblemente mejorarían el rendimiento del sistema.

Posteriormente, podríamos realizar un estudio más detallado de técnicas alternativas a las ya estudiadas, como por ejemplo el uso de, entre otros, coeficientes LPC como características de entrada del sistema, o la prueba de técnicas de normalización como pueden ser la normalización de longitud, la normalización WCCN (*Within Class Covariance Matrix)* o la aplicación de withening a los i-vectors para la mejoría del rendimiento (23) (24).

7 Referencias

1. **Alfonso Ortega, Ignacio Viñals, Antonio Miguel, Eduardo Lleida.** *The Albayzin 2016 Speaker Diarization Evaluation.*

2. **Rodríguez, Cristian Sánchez.** *Segmentation and Detection of Audio Sources in Broadcast and Telephone Audio Streams.*

3. **Mickael Rouvier, Gregor Dupuy, Paul Gay, Elie Khoury, Teva Merlin, Sylvain Meignier.** *An Open-source State-of-the-art Toolbox for Broadcast News Diarization.* s.l. : INTERSPEECH, 2013.

4. **Naranjo, Benjamín García.** *Segmentación de audio broadcast* . Enero 2016.

5. **Shum, Najim Dehak and Stephen.** *Low-dimensional speech representation based on Factor Analysis and itsapplications.*

6. **Bishop, Christopher M.** .*Pattern recognition and machine learning.* s.l. : springer, 2006.

7. **Tomi kinnunen, Haizhou Li.** *An overview of text-independent speaker recognition: From features to supervectors.*

8. **W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff** *"SVM based speaker verification using a GMM supervector kernel and NAP variability compensation,"* **..** *Toulouse, France : IEEE Int. Conf. Acoust., Speech, Signal Process., 2006, Vol. 100.*

9. **P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel.** *"Joint factor analysis versus eigenchannels in speaker recognition".* 4, s.l. : IEEE Trans. Audio, Speech, Lang. Process, May 2007, Vol. 15.

10. **Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet.** *Front-End Factor Analysis for Speaker Verification*
.

11. **Lei, Howard.** *Joint Factor Analysis (JFA) and i-vector Tutorial* .

12. **Fernández, Santiago de la Fuente.** *Análisis Factorial.*

13. **Pablo Ramírez Hereza, Javier Franco Pedroso, Joaquín González Rodríguez.** *ATVS-UAM System Description for evaluation Albayzin 2016 Speaker Diarization.*

14. **KALDI.** *[En línea] http://kaldi-asr.org.*

15. **Anguera, Xavier.** *Hidden-Markov Models (HMM) for speech Processing.*

16. **Javier Franco Pedroso, Ignacio Lopez Moreno, Doroteo T. Toledano Joaquin Rodríguez.** *ATVS-UAM System Description for the Audio Segmentation and Speaker.2010, FALA.*

17. **MATLAB.** *[En línea] https://www.mathworks.com/products/matlab.html.*

18**. Perl.** *[En línea] https://www.perl.org/.*

19. **NIST** *Rich Transciption Evaluation.***.**

20. **Claude Barras, Xuan Zhu, Sylvain Meignier, and Jean-Luc Gauvain, Member, IEEE.** *Multistage Speaker Diarization of Broadcast News . 5, SEPTEMBER 2006***.**

21. **Leeuwen, David A. van.** *"Speaker linking in large data sets".Czech Republic : in Proceedings of Odyssey 2010: The Speaker and Language Recognition Workshop, June 2010.*

22. P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam and P. Dumouchel. *PLDA for speaker verification with utterances of arbitrary duration,". s.l. : International Conference on Acustocis, 2013.*

23. Espy-Wilson, D. Garcia-Romero and C. Y. *"Analysis of ivector length normalization in speaker recognition systems," . Florence, Italy : In Proc. Interspeec, Agosto 2011.*

24. A. Hatch, S. Kajarekar, and A. Stolcke. *"Within-class covariance normalization for SVM-based speaker recognition" . Pittsburgh, Pa: in Proc. Int. Conf. Spoken Lang. Process, Sep. 2006.*

25. Wang, Shih-sian Cheng and Hsin-min. *METRIC-SEQDAC: A Hybrid Approach for Audio Segmentation . Institute of Information Science, Academia Sinica, : s.n.*

26. Mirò, Xavier Anguera. *Robust Speaker Diarization for Meetings .*

27. Sue E. Tranter, Member, IEEE and Douglas A. Reynolds, Senior Member, IEEE. *An Overview of Automatic Speaker Diarization Systems.*

28. Morancho, Gonzalo Soriano. *Diarización de Locutores en Audio Broadcast.*

29. D. Reynolds, T. F. Quatieri, and R. B. Dunn. *"Speaker verification using adapted gaussian mixture models,". Digital Signal Process, 2000, Vol. 10.*

30. Espy-Wilson, D. Garcia-Romero and C. Y. *"Analysis of ivector length normalization in speaker recognition systems," . Florence : in Proc. Interspeech 2011, Aug. 2011.*

Glosario

API	Application Programming Interface
MFCC	Mel Frequency Cepstral Coefficients
FFT	Fast Fourier Transform
CMN	Cepstral Mean Normalization
CMVN	Cepstral Mean and Variance Normalization
SAD	Speech Activity Detection
BIC	Bayesian Information Criterion
GLR	Generalized Likelihood Ratio
DCT	Discrete Cosine Transform
AHC	Agglomerative Hierarchical Clustering
HMM	Hidden Markov Model
GMM	Gaussian Mixture Model
EM	Expectation-Maximization
UBM	Universal Background Model

Anexos

A ATVS-UAM System Description for the Albayzin 2016 Speaker Diarization Evaluation

ATVS-UAM System Description for the Albayzin 2016 Speaker Diarization Evaluation

Pablo Ramirez Hereza, Javier Franco-Pedroso, and Joaquin Gonzalez-Rodriguez

ATVS - Biometric Recognition Group
Escuela Politecnica Superior, Universidad Autonoma de Madrid (Spain)
pablo.ramirezh@estudiante.uam.es
{javier.franco,joaquin.gonzalez}@uam.es
http://atvs.ii.uam.es

Abstract. This document describes the three speaker diarization systems developed by the ATVS Biometric Recognition Group, at Universidad Autonoma de Madrid (UAM), for the Albayzin 2016 Speaker Diarization Evaluation. The primary system is based on classical segmentation and clustering stages through GLR, BIC and AHC techniques, applied to MFCC features. Both contrastive systems are based on i-vectors extracted through a short-time sliding-window, resulting in a stream of i-vectors for each testing file. The first contrastive system is based on AHC directly applied to these i-vectors. The second contrastive system uses same segmentation and clustering techniques as the primary one, but applied to the stream of i-vectors.

Keywords: speaker diarization, GLR, BIC, AHC, i-vectors

1 Voice Activity Detection

Voice activity detection for the three submitted systems is based on the trajectories of the harmonics in the spectrogram, which can be used to distinguish between speech and music or noise, as these trajectories show particular an unique patterns for speech signals.

First, the amplitude of the audio signal is normalized through a 5-second length triangular sliding-window with 50% overlap, applying a variable gain inversely proportional to the square root of the energy of the signal inside the window. Then, the log-spectrogram of the signal is computed with 10 ms resolution and divided into 6 octaves with 40 logarithmic bins each. Based on this representation of the audio signal, cross-correlation values are computed for different time lags and frequency offsets, and the trajectory of the harmonics estimated from the maxima of these values, providing a score for the frame under analysis. Finally, frames are classified as speech or non-speech by comparing the scores with a threshold.

2 Feature Extraction

2.1 Primary System

For the primary system, Kaldi software [1] was used to extract one feature vector every 10 ms by means of a 20 ms Hamming sliding window (50% overlap). For each window, 20 MFCC features (including C0) were computed from 25 Mel-spaced magnitude filters over the whole available spectrum (0-8000 Hz). No channel normalization techniques were applied.

2.2 Contrastive Systems

For the contrastive systems, in-house software was used to extract one feature vector every 10 ms by means of a 20 ms Hamming sliding window (50% overlap). For each window, 19 MFCC features (without C0) were computed from 25 Mel-spaced magnitude filters over the whole available spectrum (0-8000 Hz). These features were mean-normalized, RASTA filtered and Gaussianized through a 3-second window.

3 Primary System

In the segmentation step, a 5-second length sliding-window, with a step size of 100 ms, is used to compute the distance between consecutive sets of frames in order to detect speaker change points. The features within each half of the window are modelled by a multivariate full-covariance Gaussian distribution, and the distance between the two distributions is computed through the Generalized Likelihood Ratio (GLR) metric [2]. Once the whole stream of features has been processed, the significant local maxima of the resulting distance curve are used to determine the speaker change points.

Segmentation results are refined in order to discard false alarm speaker change points through a linear clustering stage. Starting from the first speaker change point found, the ΔBIC metric [3] is computed between each pair of consecutive segments, each of which is modelled as a multivariate Gaussian with full covariance matrix. If $\Delta\text{BIC} < 0$, the speaker change point is discarded and the two segments are merged. The process is repeated until the last speaker change point.

Then, a bottom-up or Agglomerative Hierarchical Clustering (AHC) is used to merge segments from the same speaker that are not adjacent. Similarly to the previous step, ΔBIC as distance metric between clusters, modelling each cluster by means of a multivariate Gaussian with diagonal covariance matrix.

Finally, the position of speaker change points is refined by applying a Viterbi realignment. Each cluster is modelled by a 8-component full-covariance Gaussian Mixture Model (GMM) trained by means of Expectation-Maximization (EM). Then, all clusters represented by the set of GMMs are used to create a left-to-right HMM, and the Viterbi algorithm is applied to obtain the most probable sequence of clusters for the observed sequence of features.

4 Contrastive Systems

Both contrastive systems are based on the same i-vector extractor [4], which allow to obtain a stream of i-vectors for the audio files to be processed. In the contrastive system 1, these i-vectors are directly clustered giving rise to the final diarization result, similarly to the system used in [5]. In the contrastive system 2, the extracted i-vectors are used as input features for the primary system.

4.1 I-vector Extraction

In the training stage, a 1024-mixtures UBM is trained using the MFCC features belonging to the speech segments of the training dataset. Then, for every speaker segment in the training set, sufficient statistics are extracted and a total variability matrix trained for 50-dimensional subspace. Then, i-vectors are also extracted for the training speaker segments and a LDA projection matrix obtained in order to compensate the intra-speaker variability of i-vectors.

In the testing stage, similarly to [6, 7], a 1-second sliding-window is used to obtain an i-vector every 20 ms, and then projected through the LDA matrix.

4.2 Contrastive System 1

Once the stream of (compensated) i-vectors has been extracted for a testing file, they are clustered based on their cosine distance. The number of clusters is controlled by the maximum allowed distance between the i-vectors and the centroid of the cluster, which is optimized on the development dataset. The centroid of the cluster is computed as the average of the i-vectors within each cluster and it represents a candidate speaker model. Conversely to the system used in [5], no further refinement through Viterbi decoding is done.

4.3 Contrastive System 2

Contrastive system 2 is based on the same segmentation and clustering scheme as the primary system, but uses the stream of i-vectors as input, without LDA projection, instead of MFCC features. However, for this system, Viterbi realignment is not used.

5 Development Results and Timing

Table 1 shows the overall results obtained on the development dataset for the primary and contrastive systems, while table 2 shows the computational requirements in terms of CPU time for the primary system. Experiments were carried out in a machine equipped with two Xeon Quad Core E5335 microprocessors at 2.0GHz (allowing 8 simultaneous threads) and 16GB of RAM.

Table 1. *Diarization Error Rate (DER) results (in % of scored speaker time) for the primary and contrastive systems in the development dataset.*

System	Missed (%)	F. Alarm (%)	Speaker error (%)	DER (%)
Primary	3.3	5.3	18.4	26.95
Contrastive 1	4.7	6.2	22.6	33.48
Contrastive 2	3.6	5.3	27.3	36.18

Table 2. *Testing time of the different stages for the primary system (per 15-minute audio file).*

Stage	Time (minutes)
Feature extraction	0.25
Voice activity detection	5.2
Speaker diarization	1.38

Acknowledgments

This work has been supported by the Spanish Ministry of Economy and Competitiveness through the project "Redes Profundas y Modelos de Subespacios para Deteccin y Seguimiento de Locutor, Idioma y Enfermedades Degenerativas a Partir de la Voz" (TEC2015-68172-C2-1-P).

References

1. KALDI, `http://kaldi-asr.org`
2. D. Wang, R. Vogt, M. Mason, S. Sridharan (2008): Automatic Audio Segmentation Using the Generalized Likelihood Ratio. In: 2nd International Conference on Signal Processing and Communication Systems, pp. 1–5 (2008)
3. Shaobing Chen, S. and Gopalakrishnan, P.: Speaker, environment and channel change detection and clustering via the bayesian information criterion. In: DARPA Broadcast News Transcription and Understanding Workshop, Virginia, USA (1998)
4. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 19 (4), 788–798 (2010)
5. Franco-Pedroso, J., Lopez-Moreno, I., Toledano, D.T., Gonzalez-Rodriguez, J.: ATVS-UAM System Description for the Audio Segmentation and Speaker Diarization Albayzin 2010 Evaluation. In: FALA: VI Jornadas en Tecnologa del Habla and II Iberian SLTech Workshop, pp. 415-418 (2010)
6. Castaldo, F., Colibro, D., Dalmasso, E., Laface, P., Vair, C.: Stream-based Speaker Segmentation Using Speaker Factors and Eigenvoices. In: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4133–4136. Las Vegas, Nevada (2008)
7. Kenny, P., Reynolds, D., Castaldo, F.: Diarization on Telephone Conversation using Factor Analysis. IEEE Journal on Selected Topics In Signal Processing 4 (6), 1059–1070 (2010)